QUALITY ASSURANCE USING OUTLIER DETECTIONS FOR CEREBELLAR LOBULE SEGMENTATION

by

Lianrui Zuo

A thesis submitted to The Johns Hopkins University in conformity with the requirements for the degree of Master of Science in Engineering.

Baltimore, Maryland

May, 2018

Abstract

The cerebellum plays an important role in motor control and cognitive activities. In recent years, several methods have been proposed for automatic cerebellum parcellation. Usually, the segmentation accuracy is evaluated by comparing with manual delineations directly. However, such comparison is impractical in real segmentation scenarios where no manual delineations are available. What is worse, when a segmentation software fails to give an accurate result, the failed segmentation will inevitably bias further studies. Therefore, there is need for an automatic approach that can detect segmentation failures and guarantee the quality of segmentation results.

The thesis has two main focuses: evaluating and validating a new approach for cerebellar lobule segmentation and designing an automatic approach for the Quality Assurance (QA) of a cerebellar lobule segmentation pipeline. In the first part of the thesis, we formulate the task of QA and introduce several metrics for evaluating the performance of segmentation software in medical image analysis. We then evaluate a newly proposed cerebellar lobule segmentation software using the introduced metrics. Statistical results show that the segmentation software can give reliable cerebellar lob-

ule segmentation results in a reasonable amount of time while sometimes the software has segmentation failures.

The second part of the thesis focuses on automatic QA using outlier detection methods. We introduce a new approach that can automatically detect segmentation failures in a set of segmentation results. The proposed QA approach analyzes all the important processing steps of a segmentation software. In addition, the proposed method provides a general framework of QA that can be modified and applied to other image processing software. Experiments were done on two datasets including healthy controls and subjects with disease. Quantitative results show that the proposed QA method achieves both high sensitivity and high specificity in outlier detection. Qualitative results show that the method can find abnormalities in a set of segmentation results, which should give researchers clues about how their segmentation algorithms perform on a new dataset without ground truth.

Primary Reader: Dr. Jerry L. Prince

Acknowledgments

I would like to express my gratefulness to my research advisor, Dr. Jerry L. Prince, who provided tremendous amount of support and ideas on this research project. I would also like to thank him for his kind help on reviewing and editing this thesis. Over the past two years at the Image Analysis and Communications Lab (IACL), he taught me how to be a good researcher and a good man.

I would like to express a special appreciation to Aaron Carass for his useful suggestions and technical supports on this project. Furthermore, I would like to thank all the members of IACL who have given comments and encouragement during the research: Yufan He, Shuo Han, Yihao Liu, David Gomez, Blake Dewey, Chandraja Dharmana, Jeff Glaister, Jacob Reinhold, Rui Shen, Muhan Shao, Xiaokai Wang, Can Zhao, Heran Yang, and Laura Granite.

Furthermore, I would like to thank my friends Dongji Gao, Kun Qian, SzuJui Chen, Zheng Wang and my family for their consistent encouragement and trust.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The human cerebellum is involved in many crucial activities of daily life including motor control and cognitive functions [1–3]. The functions of the cerebellum are observed to be related to different cerebellar regions in clinical and experimental studies [4]. For instance, Schmahmann et al. [3] pointed out that lesions in the posterior lobe and vermis of the cerebellum have more prominent effects on behavioral changes. Wu et al. [5] observed that patients with Parkinson's disease have different extents of atrophy in different cerebellar regions.

The human cerebellum is a structure attached at the bottom of the brain [6]. In [7], Schmahmann et al. divided the human cerebellum into 11 structures named cerebellar lobule I to X, and the corpus modulare. The 10 cerebellar lobules are separated and identified by cerebellar fissures, as shown in Fig. 1.1. In each lobule, there are at least two substructures corresponding to the left and right hemisphere.

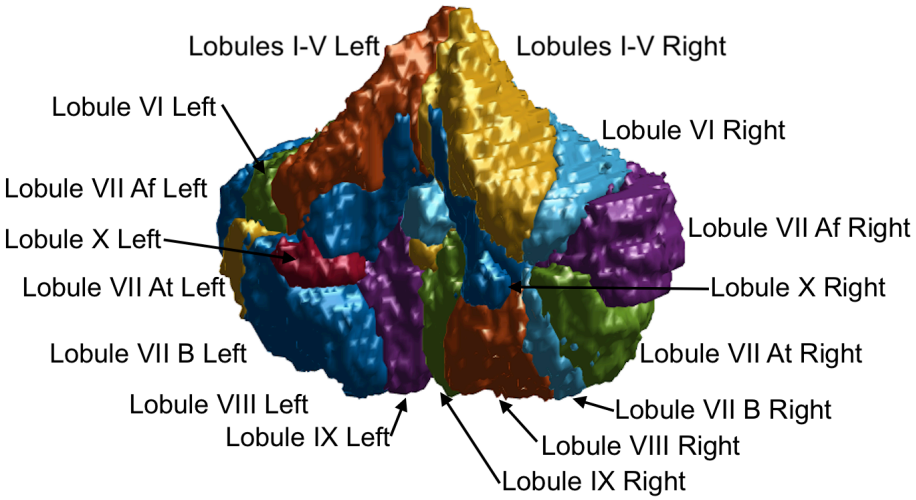


Figure 1.1: Visualization of the human cerebellum in a coronal view.

In lobules VI through X, there is a vermis structure at the "midline" of the lobule [7].

To better understand the anatomical structure of the human cerebellum, especially for patients with cerebellum-related disease, researchers have proposed several cerebellum segmentation algorithms. Diedrichsen [8] proposed a spatially unbiased atlas template of the human cerebellum (SUIT), which can be used to do single-atlas segmentation for human cerebellum. Bogovic et al. [9] proposed ACCLAIM (Automatic Classification of Cerebellar Lobules Algorithm using Implicit Multi-boundary evolution), which segments the cerebellum using geometric deformable model followed by a random forest classifier. Yang et al. [1] proposed the cerebellar lobule segmentation software and the software has been shown to have the state-of-the-art performance in cerebellar lobule segmentation. These segmentation softwares facilitate the stud-

ies on the morphology of the cerebellum, especially in large datasets, since manually delineating a human cerebellum in an MR image is extremely time-consuming and requires professional knowledge [10].

Sometimes the segmentation software may fail to give an accurate and reliable result for various reasons such as the low quality of MR images, severe cerebellar atrophy, or even misoperating the software. Obviously, the segmentation failures will inevitably bias the study of the cerebellum. One way of solving this problem is to conduct a Quality Assurance (QA) process. QA is a category of activities that seeks to ensure a consistent and optimal performance [11]. Among all approaches, one straightforward and efficient way of conducting a QA for medical image analysis is to do an outlier detection. Usually, outlier detection is conducted by manual inspection. Grubbs [12] stated that an outlying observation, or an outlier, is the one that appears to deviate markedly from other members of the sample in which it occurs.

Although QA is highly needed in various fields, research on QA for medical image analysis is limited. In most scenarios of image processing and analysis, to evaluate and validate a software, researchers compare image processing results with the truth created by human raters, and they have no idea about how the algorithm will perform on new data without ground truth. To conduct a higher-level analysis such as disease diagnosis based on segmentation, researchers must manually detect segmentation outliers to guarantee a reliable diagnosing specificity and sensitivity. Therefore, there is need for an automatic outlier detection approach that can efficiently and accurately

find outlying subjects.

In this thesis, we focus on two main topics: evaluating and validating a new segmentation software and a new approach to do QA using automatic outlier detection for medical image segmentation algorithms. The thesis is based on a specific segmentation software for cerebellar lobule parcellation. However, the introduced idea and framework can be extended and applied on other segmentation algorithms of medical image analysis.

In this chapter, the main contributions and the thesis organization are described.

1.1 Thesis contributions

There are two main contributions in this thesis:

(1) Evaluation and validation of a segmentation algorithm.

We validated a new segmentation software for cerebellar lobule segmentation proposed by Yang et al. [1] in 2016. The evaluation and validation work was done from the aspect of *accuracy* and *efficiency* of the algorithm. Step-wise analysis by both Wilcoxon test and human rater visualization was conducted to validate the rationality of several important processing steps. Feature importance in algorithm training was also studied. Results show that the cerebellar lobule segmentation software can give reliable segmentations for most cerebellar structures. However, the software fails to produce accurate segmentations

occasionally, yielding segmentation outliers.

(2) A new approach of QA using outlier detection for medical image segmentation.

We proposed a new approach of outlier detection for medical image segmentation. The method can automatically detect segmentation outliers without comparing with a true segmentation. The outlier detection software first analyzes the output of each image processing step sequentially, then uses a Hidden Markov Model (HMM) [13] to give a global assessment on all procedures of a segmentation software. To our knowledge, this is the first work using HMMs to do outlier detection in medical image analysis. Results show that the outlier detection approach achieves both high sensitivity and specificity in detecting segmentation failures.

1.2 Thesis organization

The thesis is organized as follows.

In Chapter 2, we introduce the background of this thesis. We first do a brief literature review on current methods of cerebellar lobule segmentation. In Section 2.2, we formulate the QA task by introducing several important definitions. The challenges in QA for medical image analysis are also introduced in this section. At the end of this chapter, we briefly describe some relevant techniques (e.g., HMM)

that are used in our outlier detection approach.

In Chapter 3, we first introduce the cerebellar lobule segmentation software by specifying its inputs and outputs and then describe several important steps of the segmentation software. Next, we present a quantitative evaluation of the segmentation software. The evaluation was done in both segmentation accuracy and segmentation efficiency. Statistical tests were done on several metrics such as Dice coefficient [14] and Hausdorff distance. Qualitative evaluation of the segmentation is also provided. Finally, we categorized the segmentation results into four categories, which we then used in our outlier detection approach.

In Chapter 4, we describe the proposed outlier detection approach in detail. We first provide a general outlier detection framework that mainly focuses on the input/output sequence instead of the final output itself. We then describe an approach to train the proposed method using a limited number of subjects. In Section 4.3, a set of experiments were done on multiple datasets. We first studied the outlier detection performance on 15 subjects which have manual delineations. At the end of this section, we did an experiment on the so-called Tomacco dataset, and qualitatively evaluated the performance of the outlier detection method.

In the final chapter, we summarize the conclusions of the thesis, and point out several directions for future work.

Chapter 2

Background

2.1 Methods for cerebellar lobule segmentation

Since the topic of this thesis is outlier detection for cerebellar lobule segmentation, we give a brief literature review on the current methods for cerebellar lobule segmentation in this section. Although image segmentation techniques have been greatly developed in the past decades, consistently and accurately segmenting the human cerebellum still remains challenging [1].

Most methods in cerebellum segmentation were proposed in the past 15 years. In 2002, Fischl et al. [15] proposed a whole brain segmentation approach, FreeSurfer, in which the cerebellum is segmented into White Matter (WM) and Gray Matter

(GM) as part of its whole brain segmentation result. Methods focusing on cerebellum parcellation are mainly atlas-based. In 2006, Diedrichsen [8] proposed a spatially unbiased atlas template of the human cerebellum (SUIT), which can be used to do single-atlas segmentation for human cerebellum. He then extended SUIT to a probabilistic cerebellar atlas in 2009 [16]. In 2013, Bogovic et al. [9] proposed ACCLAIM, which uses a geometric deformable model and a random forest classifier to do cerebellum parcellation. ACCLAIM was shown to have better segmentation accuracy than SUIT. In 2016, Yang et al. [1] proposed a cerebellar lobule segmentation software using a multi-atlas based segmentation method followed by a graph cut [17]. The method was reported to have state-of-the-art performance in segmentation accuracy. In 2017, Romero et al. [18] developed a new cerebellum lobule segmentation method (CERES), which adopts an optimized label fusion approach and multi-atlas patch-based segmentation method to do cerebellar lobule segmentation. The method achieves both high accuracy and high speed.

2.2 QA in medical image segmentation

QA for medical image processing and analysis algorithms is an application-based activity; in different application scenarios, QA can have distinct procedures and requirements. Therefore, it is essential for every QA researcher to specify an *application domain* before conducting QA. Udupa et al. [19] defined an *application domain*

$\langle T, B, P \rangle$ of the QA in medical image analysis by specifying three factors:

T : A task. Example: image segmentation.

B : A body region. Example: human brain.

P : An imaging protocol. Example: MR imaging.

In this thesis, we introduce our QA framework by specifying the task, T_1, to be image segmentation, while our region of interest, B_1, is the human cerebellum, and the imaging protocol, P_1, is MR imaging. However, the proposed framework can be extended and applied in other *application domains* with little modification.

Literature on QA for medical image segmentation algorithms is limited [20]. There are mainly two aspects of evaluating the quality of a medical image processing algorithm: from the algorithm itself (e.g., time complexity) and by evaluating the output of the algorithm.

The output of an image segmentation algorithm is often a mask image which labels the regions of interest. In medical image segmentation, a region to be segmented often refers to a specific structure that carries useful anatomical information, such as a brain ventricle or a cerebellar lobule. A reliable segmentation could guide researchers to make higher-level observations efficiently. For example, with the help of image segmentation, researchers can study the effects of cerebellar atrophy and track the progress of disease by doing shape analysis on a set of images [21]. From this aspect, the quality of a segmentation result can be understood as how well the result fulfills

its application task. On the other hand, a failed segmentation may significantly bias the higher-level study.

2.2.1 QA using outlier detection

One efficient and powerful approach to evaluating the quality of the outputs of a segmentation algorithm is conducting outlier detection. Outlier detection is an essential procedure in many QA activities, since the presence of segmentation outliers is highly related to the performance of a segmentation software. Considering there is no universally accepted definition of an outlier, we take the definition of Grubbs [12], who defined an outlier to be one that appears to deviate markedly from other members of the sample in which it occurs. This definition provides a statistical approach for treating outliers. In the following chapters, we will further narrow our focus by only considering QA using outlier detection in *application domain* $\langle T_1, B_1, P_1 \rangle$.

In medical image segmentation, many researchers evaluate the segmentation results of their algorithms by directly comparing them with manual delineations. Their assumption is that manual delineations represent perfect segmentations. This is a "one-to-one" comparison in image space (i.e., a high-dimensional space that contains all voxels). However, for most subjects, manual delineations are not available. In such cases, a "one-to-one" comparison is not available, and human raters are needed to classify bad segmentation results. This can be very time-consuming in a large dataset and requires professional knowledge.

Some researchers proposed automatic methods to find outliers from a set of segmentation results [22,23]. The basic assumption is that there are anatomical similarities in the segmented structure among populations; successful segmentations carry these similarities, while failed segmentations do not. Therefore, there are two essential factors that dominate the outlier detection task: the choice of image features which can represent the anatomical similarities and the choice of models that map features to QA decisions. Mathematically, outlier detection can be understood as first looking for a function $f(\cdot)$ that can map a testing image $\boldsymbol{I}$ to a point $\boldsymbol{u}$ in a feature space, then looking for a function $g(\cdot)$ that can make QA decisions Q based on the feature vector $\boldsymbol{u}$, i.e., $Q = g(f(\boldsymbol{I}))$. It is worth pointing out that it is not necessary to find $g(\cdot)$ and $f(\cdot)$ separately. In some outlier detection methods, the software directly finds a regression from the image $\boldsymbol{I}$ to a decision Q. In our case, the image vector $\boldsymbol{I}$ can be any image format, such as a segmentation result or even the original MR image. It is also admissible to have multiple $\boldsymbol{I}s$ as inputs to make a QA decision based on sequential data. The decision Q refers to either a binary value indicating success/failure or a rating score representing the confidence of a success/failure.

Fig. 2.1 shows the *volume* of Lobule VII right Af of 15 subjects. We can see that there is an outlying observation in Subject #2, as indicated by a black arrow. We inspected this segmentation result and compared it to a corresponding manual delineation. As shown in Fig. 2.2(a), the segmentation software failed to capture the correct structure of the cerebellar lobule, which caused a segmentation outlier. In

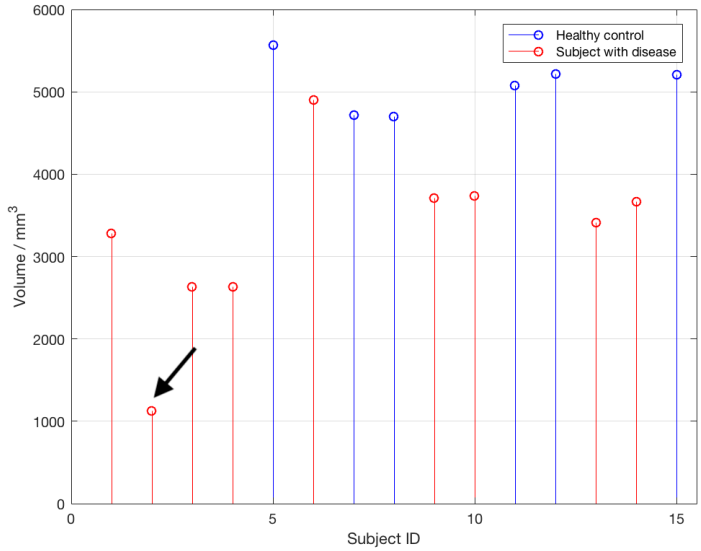
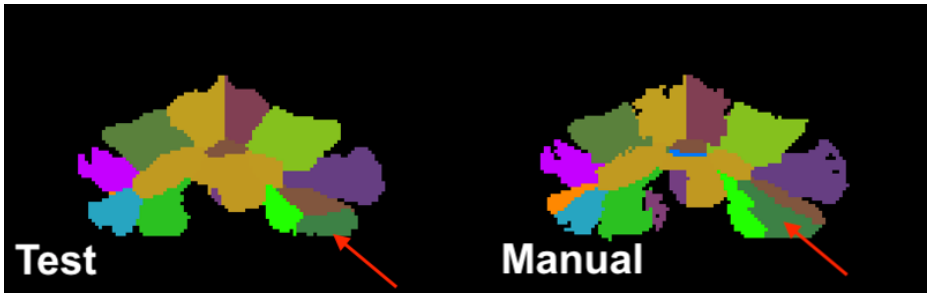


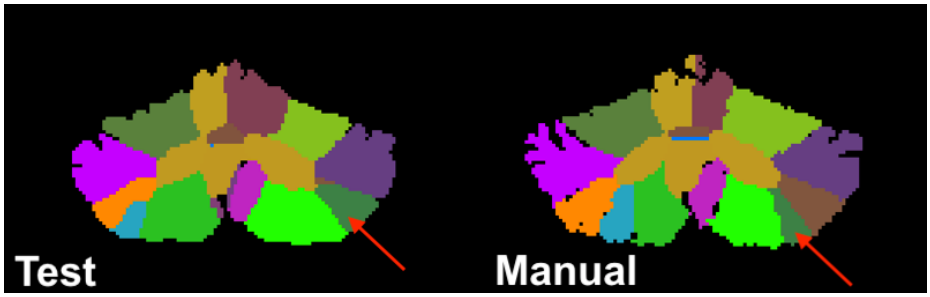
Figure 2.1: A visualization of an outlier in an 1-D feature space.

this example, the label *volume* serves as a feature that assists us to find an outlying observation in an 1-D feature space. However, after manual inspection on all 15 subjects, we found that the segmentation result of Subject #11 is also an outlier, as shown in Fig. 2.2(b). Unfortunately, according to the *volume* information, we cannot tell whether the segmentation of Subject #11 is an outlier. Another interesting observation from Fig. 2.1 is that heathy subjects and subjects with cerebellar disease have different patterns of *volumes*; healthy subjects have larger label *volumes* on average than subjects with disease. This phenomenon is categorized as population variation, as we shall explain in the next section.

Work in finding feature functions $f(\cdot)$ (i.e., mapping the original image to a feature space) is very ad-hoc—the image features that $f(\cdot)$ finds only function on a



(a)



(b)

Figure 2.2: A visual comparison of **(a)** Subject #2 and **(b)** Subject #11 with corresponding manual delineations.

specific algorithm. Li et al.'s [22] outlier detection method for cerebellar peduncle segmentation is only applicable on a small kind of segmentation software since it requires features from DTI images and whole brain segmentations, which are not often available. There are also trade-offs between feature generality and feature sensitivity. On the one hand, sensitive features may only available in a small portion of cases. On the other hand, general features may not have good performance in some specific tasks.

Looking for a decision function $g(\cdot)$ can be understood as either a classification or a regression problem. Although the work in QA is limited, much work has been

done in designing regression functions and classifiers, and it can be applied to finding outliers in a feature space. Famous approaches include K-means [24] and Support Vector Machine (SVM) [25]. In some cases, dimensionality reduction methods are needed to reduce unnecessary variations and prevent potential over-fitting.

2.2.2 Challenges in outlier detection

There are several challenges in outlier detection that prevent researchers from conducting accurate QA in medical image analysis.

(1) The chicken-and-egg problem.

There is a question that has been asked several times: if there is a method to automatically detect segmentation outliers, why not use the method to do a better segmentation? Intuitively, this is a chicken-and-egg problem. Under our definition of an outlier in Chapter 1, the (statistically) better a segmentation algorithm is, the harder for a QA algorithm to detect outliers. To be concrete, an outlier of a good segmentation software might resemble more desired anatomical information than an inlier of a poor segmentation software. Therefore, as the quality of segmentations becomes better, it is harder to perform an accurate QA using outlier detection.

In recent years, based on this idea, researchers have proposed Generative Adversarial Nets (GAN) [26], a deep learning approach for image segmentation. In a GAN, there is a generator and a discriminator. The generator and the discriminator are competing with each other; the generator aims to produce a result that can fool the

discriminator, while the discriminator tries to distinguish which is good and which is bad.

Although outlier detection and image segmentation encourage each other to a better performance, which seems like a chicken-and-egg problem, there are distinctions between them. Technically speaking, image segmentation aims to conduct a pixel/voxel-wise decision that decides which class each pixel/voxel belongs to. In outlier detection, researchers not only have pixel/voxel-wise information, but also global information (e.g., the shape of a segmentation result, the intensity distribution within a segmented region, etc.). In addition, an outlier detection algorithm aims to produce a subject-wise decision, which indicates the quality of each segmentation result.

In [27], the author categorized the labeling problems (LP) in computer vision into 4 categories:

LP1 : Regular sites with continuous labels (e.g., image smoothing).

LP2 : Regular sites with discrete labels (e.g., image segmentation).

LP3 : Irregular sites with discrete labels (e.g., object matching).

LP4 : Irregular sites with continuous labels (e.g., scene recognition).

Obviously, most segmentation work in medical image analysis is *LP2* while outlier detection, whose objective is to extract information from segmented images and tell researchers whether the segmentation is trustworthy or not, is *LP3*.

(2) Variation in populations.

Except for segmentation outliers, which can be understood as an extrinsic factor that influences a segmentation result, there are many intrinsic factors that can greatly affect the outputs of a segmentation algorithm. For example, the deformation in brain structure resulting from aging or disease may cause a segmentation outlier even though it successfully captures the brain structure. The whole set of segmentation results can be divided by success/failure, young/old, healthy/disease, or other factors. Such factors make outlier detection an even harder task, since it is difficult to decide a dominate factor that causes an abnormal observation. In recent years, Wachinger et al. [28] proposed a method to study the latent processes which govern the observations. Their method successfully separate the different effects of aging and disease on a certain set of brain structures, which shows promise in our problem. It is desired to be able to separate the dominant factor of causing an abnormal observation.

(3) Limited training data.

As mentioned above, the outlier detection task is *LP3* and it often needs a subject-wise decision. In medical image analysis, providing manual delineations and diagnose is extremely expensive. Therefore, training data in medical image processing is less common than general computer vision. What is worse, there are countless numbers of failure cases for software, which makes capturing all the modes of failure impractical.

To address this issue, Varol et al. [29] proposed clustering methods aiming at finding inliers and excluding outliers via convex polytopes.

2.3 Related techniques

This section provides some background knowledge about image features and feature modeling methods. These methods are used in our proposed QA approach that is described in Chapter 4.

2.3.1 Scale invariant feature transformation

The Scale Invariant Feature Transformation (SIFT), originally proposed by Lowe in 1999 [30], is a gradient-based image feature transformation. It aims at finding a transformation $f(\cdot)$ that maps the original image to a feature space constructed by histograms of the image gradient. SIFT has been used in various applications, such as scene matching [31], gesture recognition [32], and brain structure analysis [33]. Lowe divided SIFT into four main steps: extrema detection in scale space, localizing critical points, determining dominant directions of each point, and critical points description. Since the main focus of this thesis is not image feature representation, we only give a brief description of Lowe's method in this section.

The SIFT method aims at finding and describing critical points at different scales. To construct a scale space, the original image should first be blurred by a Gaussian

kernel. Then the scale space $L(x, y, \sigma)$ of a 2D image $I(x, y)$ is defined as the convolution of the Gaussian kernel $G(x, y, \sigma)$ with the original image

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y).$$

Mikolajczyk et al. [34] found that the minima and maxima of normalized Laplacian of Gaussian (LoG) function $\sigma^2 \nabla^2 G$ can produce stable image features. Lowe used a more efficient operator, Difference of Gaussian (DoG), to approximate the LoG, which is defined as

$$\text{DoG}(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma).$$

The final critical points are selected according to the extrema of DoG in a constructed scale pyramid after eliminating some false alarms (i.e., non-critical points that are mistakenly selected).

After locating critical points, SIFT then constructs a gradient histogram of each critical point c_i to find a dominant gradient direction. The gradient of all adjacent pixels of c_i within a radius is weighted by the Gaussian kernel at the current scale. In histogram, the bin that has the largest number of counts represents the dominant gradient direction of critical point c_i. The step of finding a dominant gradient direction guarantees a rotation invariant property of SIFT descriptors.

After all the previous steps, every critical point c_i has three parameters: location, scale, and a dominant direction. The final step of SIFT is to extract a gradient histogram at each 4×4 image block. In contrast to previous gradient histogram,

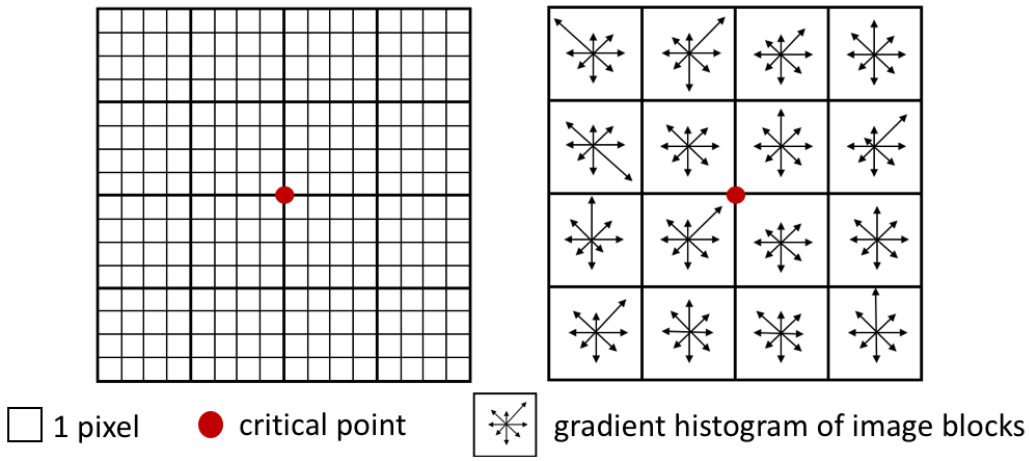


Figure 2.3: An illustration of extracting SIFT descriptors after finding critical points.

in this step, the gradient direction of every point is calculated with respect to the dominant direction. Fig. 2.3 is an illustration of the final step of SIFT extraction. In 2D, the SIFT descriptor of a every critical point is a 128-dimension vector when using a 4×4 block to construct the gradient histogram.

2.3.2 Hidden Markov model

The mathematical representation of an HMM was originally developed by Baum et al. [13] in 1966. As shown in Fig. 2.4, each node represents a random variable. The value of the random variable cannot be observed directly (it is hidden), but there are some observable variables that are related to each hidden random variable. One goal of an HMM is to extract unobservable information based on observed data. In general, an HMM λ can be parameterized by three parameters (i.e., $\lambda = [\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}]$). In Chapter 4, we will use the following notations to describe our method.

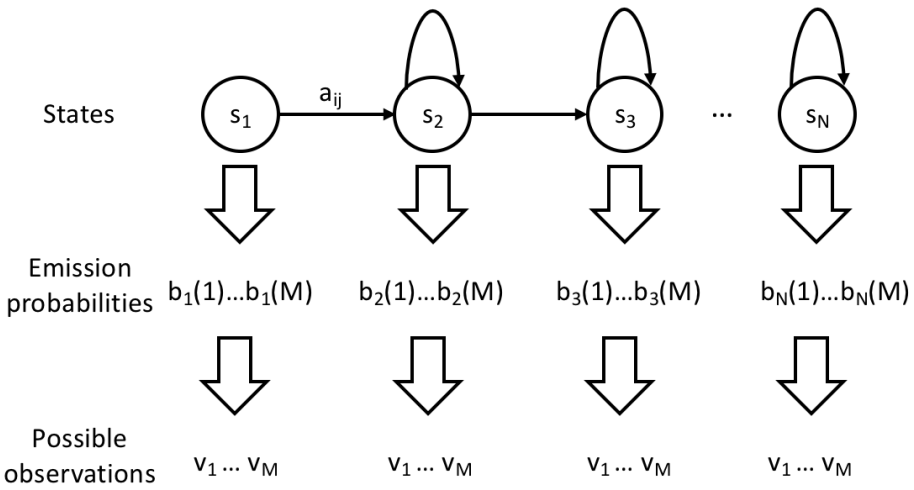


Figure 2.4: The structure of an N-step HMM with a total of M possible observations.

- $\boldsymbol{s} = [s_1, s_2, ..., s_N]$ contains N possible states of a given HMM.
- $\boldsymbol{v} = [v_1, v_2, ..., v_M]$ contains M possible observations of all possible states.
- A hidden state sequence $\boldsymbol{q}$ of length T and its corresponding observation sequence $\boldsymbol{o}$ are denoted: $\boldsymbol{q} = [q_1, q_2, ..., q_T]$ and $\boldsymbol{o} = [o_1, o_2, ..., o_T]$.
- $\boldsymbol{A} = [a_{ij}]$ is the transition probability, where a_{ij} denotes the probability of state s_j following state s_i, and can be written as $a_{ij} = P(q_t = s_j | q_{t-1} = s_i)$.
- $\boldsymbol{B} = [b_i(k)]$ is the emission probability, where $b_i(k)$ denotes the probability of observing an observation v_k from state s_i, which can be written as $b_i(k) = P(o_t = v_k | q_t = s_i)$.
- $\boldsymbol{\pi} = [\pi_1, \pi_2, ..., \pi_N]$ is the prior probability, and $\pi_i = P(q_1 = s_i)$.

2.4 Chapter summary

In this chapter, we first did a brief literature review on automatic cerebellar lobule parcellation methods. We then descirbed the background of the QA work in medical image segmentation by introducing the definition of *application domain*, and pointing out challenges in QA. Finally, we provided some related techniques that are used in Chapter 4.

Chapter 3

Validation of Cerebellar Lobule Segmentation Software

The quality assurance work of this thesis is based on the cerebellar lobule segmentation software using graph cuts proposed by Yang et al. [1] in 2016. In Section 3.1, we give a brief description of the segmentation software by demonstrating its inputs and outputs and the role of each processing step. In Section 3.2, we introduce a set of experiments designed to validate the segmentation software statistically. The validation work was done by comparing segmentation results with manual delineations. We chose multiple metrics such as Dice coefficient [14] and Hausdorff distance to make our validation work more persuasive.

In the end of this chapter, we categorize different kinds of segmentation failures, which we will further explore in Chapter 4. Results show that the software can give

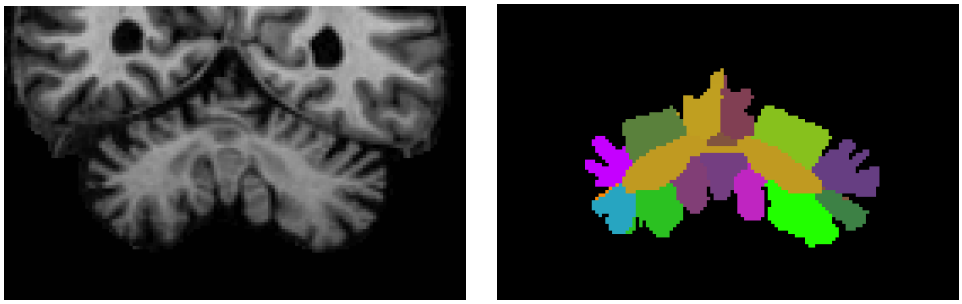


Figure 3.1: The output of Yang et al.'s segmentation software.

reasonable and reliable segmentation results in most cases.

3.1 Software description

As shown in Fig. 3.1, the output of the software is a 3D mask image with 22 non-overlapping labels. Each label corresponds to an anatomical structure of the human cerebellum. The segmentation method was reported to have state-of-the-art performance [1], and has been used by researchers around the world.

The software requires FreeSurfer [15] to preprocess the original MR image. There are two roles of FreeSurfer. First, the original MR image needs to be transformed to MNI space [35], then skull stripped and intensity corrected by FreeSurfer. In addition, FreeSurfer can provide an initial segmentation result of the cerebellum (i.e., WM and GM regions of the cerebellum). Both the FreeSurfer processed intensity image and the generated GM/WM labels serve as inputs to the cerebellar lobule segmentation software.

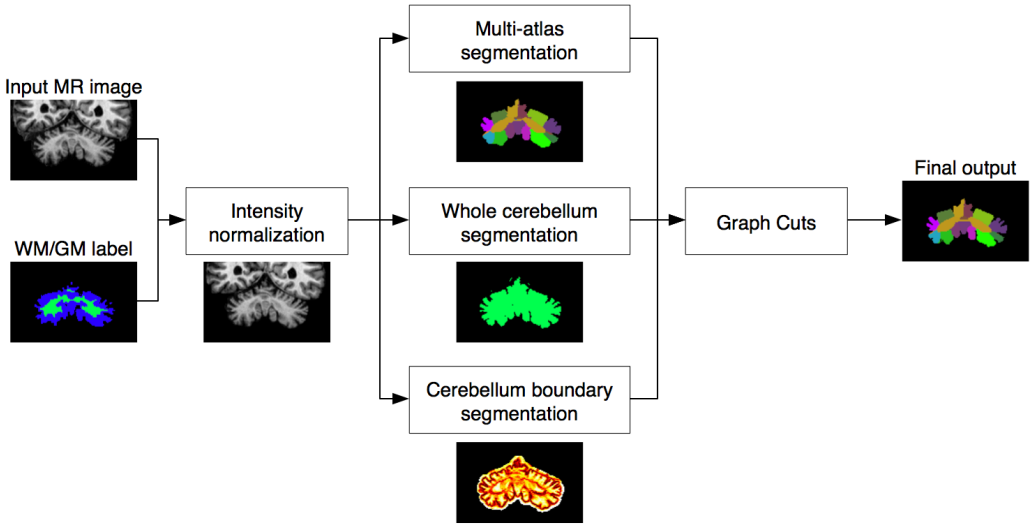


Figure 3.2: Diagram of Yang et al.'s [1] segmentation software.

Fig. 3.2 is a diagram of the segmentation software. The diagram is mainly based on the framework of the graph cut method [36, 37]. The graph cut algorithm aims to find an optimal image segmentation result by minimizing an energy function. A typical graph cut energy function has a region term and a boundary term [1]. In Yang et al.'s software, the region term refers to the region of the whole cerebellum and each cerebellar lobule. The segmentation of cerebellum boundaries and the whole cerebellum region are obtained by a pretrained random forest (RF) [38] classifier, while the multi-atlas segmentation result comes from the non-local STAPLE (NL-STAPLE) algorithm [39].

An RF classifier is a combination of decision trees such that each tree depends on the values of an independently sampled random vector [38]. Implementing an RF as a classifier has two main steps: model training and prediction. The training step

requires a sampling matrix, of which each row is a sample in the region of interest (ROI), and each column is a feature component of the corresponding sample. In addition, a true label of each sample is required to establish decision trees in the forest. The output of the training step is a trained model, and it can be used in testing phase to predict the class of each testing sample.

A multi-atlas labeling method uses image registration software to register a series of labeled images to a target testing image, then it performs a label fusion step to make a decision on all candidate labels at each voxel. NL-STAPLE aims to find a probabilistic map to achieve the label fusion goal.

There are 15 subjects with manual delineations that can be used as training data of the segmentation software. As mentioned above, the voxels of each subject are sampled randomly, and the corresponding image features along with the true labels are extracted to train an RF classifier. The trained RF model for whole cerebellum segmentation has two voxel classes—cerebellum region and non-cerebellum region, while the model for cerebellum boundaries has three tissue classes: outside cerebellum, inside cerebellum, and cerebellum boundaries. In each tissue class, an 11-dimensional feature vector is extracted. The 11 features are *intensity*, *gradient in 3D*, *magnitude of gradient*, *location in 3D*, *Euclidian distance to the corpus medullare*, and *3 components of the Hessian matrix*. As we shall explore in Section 3.2, these features are of different importance and they together contribute greatly to the final segmentation result. In NL-STAPLE, the 15 subjects and their manual delineations are registered

to a test image. The multi-atlas segmentation result provides a coarse segmentation for each lobule in further processing steps.

3.2 Statistical analysis

To evaluate a segmentation software, there are three factors that should be considered – *precision*, *accuracy*, and *efficiency* [19]. In specific, *precision* refers to the repeatability of a segmentation that takes all subjective factors into consideration, such as the patient's position in a scanner. *Accuracy* represents how well a segmentation result agrees the true segmentation. *Efficiency* often refers to the total time required to produce a segmentation result. In this section, we explore the *accuracy* and *efficiency* of the segmentation software via statistical analysis.

3.2.1 Metrics of evaluating accuracy

In most application scenarios, researchers evaluate their segmentation results by directly comparing them with manual delineations. A large number of papers [40–42] in medical image segmentation use the Dice coefficient [14] as a measurement to validate segmentation performance. Udupa et al. [19] use *True Positive (TP)* and *True Negative (TN)* to assess lesion segmentation algorithms. In [43], the authors used multiple metrics including Dice coefficient and average surface distance (ASD) as a combined scoring system to evaluate liver segmentation in the MICCAI 2007 chal-

lenge. Yeghiazaryan and Voiculescu [44] categorized the evaluation methods into two main groups—overlap based methods and distance based methods. In this section, metrics of evaluating segmentation accuracy are introduced and then in the next a few sections we use the metrics to assess the cerebellar lobule segmentation software.

(1) Dice coeffcient

Dice coefficient is an overlap based measure. It measures the overlap between two sets (e.g., mask images) with respect to the total size of the two sets. Suppose there are two binary images A and B with intensity value 1 representing the structure to be segmented and 0 elsewhere, the Dice coefficient is defined as

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}.$$

The Dice coefficient has many good properties. It has a range $[0, 1]$, with 0 for a complete mislabeling, and 1 for a perfect segmentation. Also, it penalizes both under-segmentation (i.e., a segmentation that is smaller than the ground truth) and over-segmentation (i.e., a segmentation that is larger than the ground truth). However, as pointed in [42], Dice is a simple metric and is sometimes not able to distinguish complicated algorithm behavior.

(2) Hausdorff distance

The Hausdorff distance is a distance-based metric which treats each segmentation result as a surface, and it calculates the largest distance between the surface of a segmentation result and that of the ground truth. It overcomes some dis-

advantages of overlap based methods. For example, overlap based methods only consider overlapping regions and have no extra penalty on label separation (i.e., a connected structure that is segmented into two separate regions). The Hausdorff distance of two sets A and B is defined as

$$d_H(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b), \sup_{b\in B}\inf_{a\in A} d(a,b)\right\},$$

where $d(a,b)$ is the Euclidian distance between point a and b, and sup and inf refer to supremum and infimum, respectively.

3.2.2 Assessment on segmentation accuracy

The images used to evaluate the segmentation results are T1-weighted Magnetization Prepared Rapid Gradient Echo (MPRAGE) images acquired on a 3.0 T scanner (Intera, Philips Medical Systems, Netherlands). The resolution of the scan is 0.828 mm $\times$ 0.828 mm $\times$ 1.1 mm. There are 15 subjects that have manual delineations, 6 of them are healthy controls and the others are patients with different kinds of cerebellar disease. Before applying the segmentation software, the original MR images were preprocessed by FreeSurfer and registered to MNI space to obtain an isotropic resolution of 1 mm^3.

We did a leave-one-out experiment on the 15 subjects that have manual delineations. As shown in Fig. 3.2, there are three important steps before giving a final segmentation result: NL-STAPLE, NL-STAPLE corrected by an RF (NL-RF), and a

Graph Cut (GC). The outputs of all the three steps are mask images containing 22 non-overlapping labels. Both overlap-based and distance-based methods introduced in Section 3.2.1 were calculated to evaluate the segmentation accuracy of each label. As shown in Figs. 3.3 and 3.4, the software can produce good segmentation results in most labels with Dice coefficient higher than 0.65. However, 15 of the 22 labels have segmentation outliers in terms of Dice coefficient, and 13 labels have segmentation outliers in terms of Hausdorff distance. In some small and thin cerebellar structures such as Lobule X Left, Lobule X Right, and Lobule VII Vermis, the software has relative low Dice coefficient. In such cases, automatic quality assurance are needed to detect outliers and guarantee reasonable results.

From Figs. 3.3 and 3.4, we can observe that the Dice coefficient and Hausdorff distance have distinct behaviors on different cerebellar structures. In small cerebellar lobules, such as the vermis lobules and lobule X, the Dice coefficient shows instability and indicates poor performance while the Hausdorff distance of these small lobules is still low. Another interesting observation is that the Hausdorff distance of Lobule Group I-V Right has three prominent outliers although all the 15 subjects have relative high Dice coefficient. One reason for such outliers is that Hausdorff distance is sensitive to protuberance.

From Figs. 3.3 and 3.4, we cannot see significant difference in the distribution of either the Dice coefficient or the Hausdorff distance of the three processing steps. We conducted a one-side paired Wilcoxon test [45] on any two combinations of the

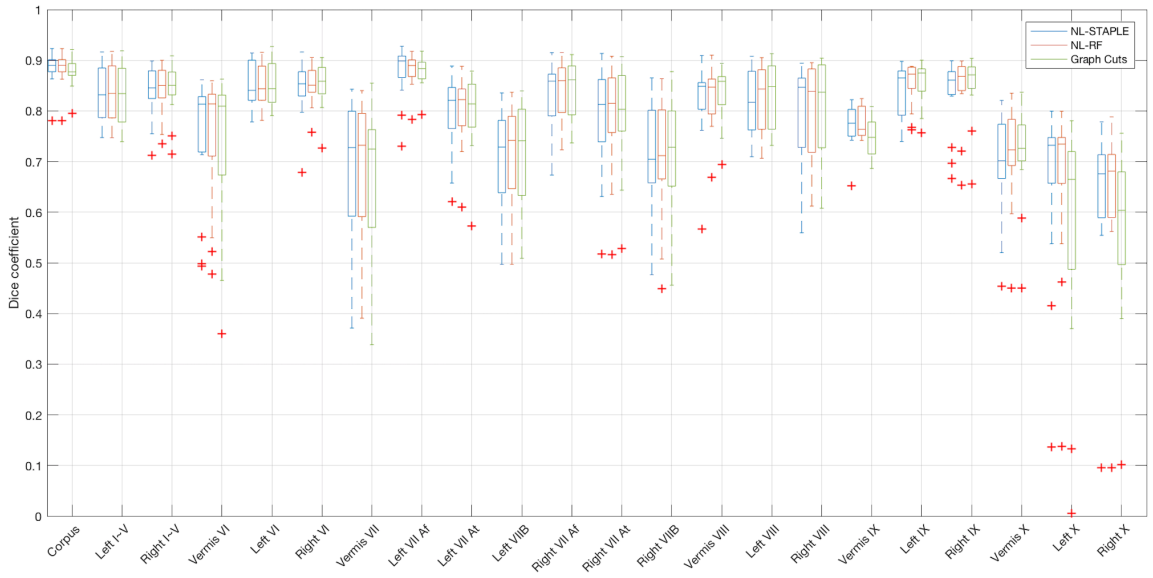


Figure 3.3: Boxplot of Dice coefficient of 15 subjects in a leave-one-out experiment.

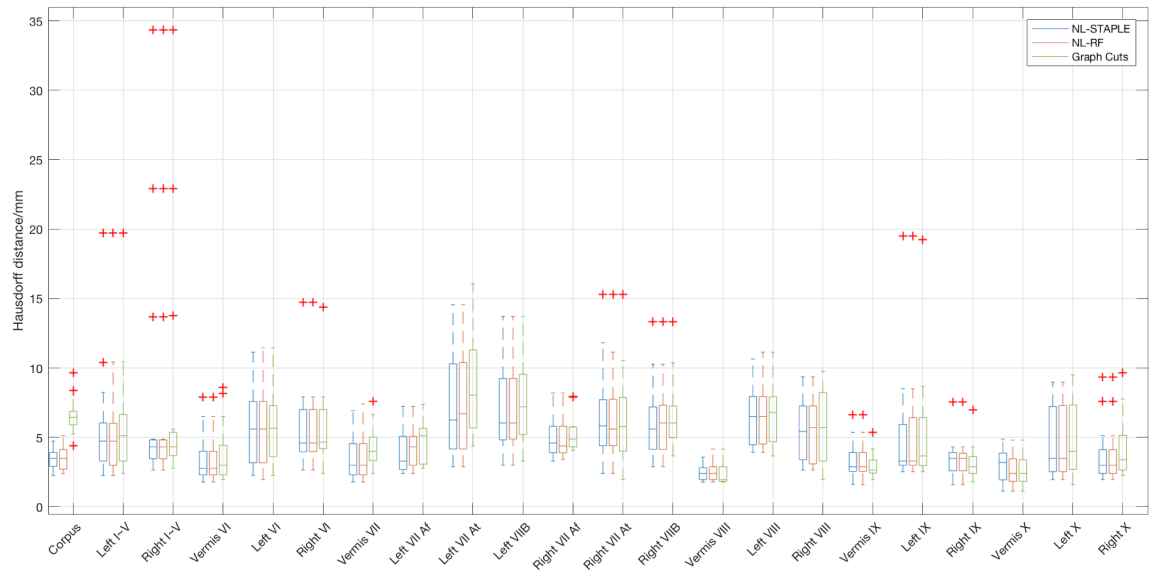


Figure 3.4: Boxplot of Hausdorff distance of 15 subjects in a leave-one-out experiment.

three steps. The purpose of the Wilcoxon test is to examine whether there is significant improvement in terms of Dice coefficient or Hausdorff distance of the three

processing steps. It is worth noting that the null hypothesis H_0 of the Wilcoxon tests on the Dice coefficient and the Hausdorff distance is different, since a higher Dice indicates better performance while higher Hausdorff distance is poorer performance. Tables 3.1 and 3.2 give p-values of the one-side Wilcoxon test on each of the 22 labels. The bold numbers indicate rejecting H_0 at a significant level $\alpha = 0.05$. In most cases, we cannot reject the null hypothesis H_0, which assumes that the later processing step (i.e., NL-RF or GC) does not produce a better result. Another observation is that there are several labels with similar p-values in the Wilcoxon test. This is because the Hausdorff distance has little change after NL-RF and GC, which can also be observed in Fig. 3.4. Because of this phenomenon, in the following sections, we only use the Dice coefficient to evaluate the segmentation result.

Unfortunately, neither boxplots nor Wilcoxon tests indicate that RF correction or GC after NL-STAPLE produces a better result in terms of Dice coefficient or Hausdorff distance. Therefore, we qualitatively compared the outputs of the three steps by manual visualization. As shown in Fig. 3.5, the output images of the three steps have almost the same Dice coefficient and Hausdorff distance, but the NL-STAPLE and GC results look better than NL-RF, since they have relative smooth boundary without bumps and holes. The GC results of an SCA6 patient shows more agreement with manual delineation. It is also worth pointing out that conducting the GC process may have negative effects on small cerebellar structures, such as Lobule X, which can also be observed from Fig. 3.3.

Table 3.1: p-value of one-side paired Wilcoxon test on Dice coefficient. H_0: there is no significant improvement in the later processing step.

	CM	**I-V L**	**I-V R**	**VI Ver**	**VI L**
NL vs NLRF	0.4452	**0.0034**	**0.0042**	0.8853	0.4890
NL vs GC	0.9966	0.6401	0.2807	0.7729	0.3394
NLRF vs GC	0.9958	0.9723	0.7378	0.8853	0.2997

VI R	**VII Ver**	**VII Af L**	**VII At L**	**VII B L**	**VII Af R**
0.1651	0.8486	0.8053	0.4670	0.1514	0.0677
0.0535	0.9584	0.8738	0.6807	0.2444	0.0844
0.2271	0.9681	0.9156	0.7556	0.5110	0.7894

VII At R	**VII B R**	**VIII Ver**	**VIII L**	**VIII R**	**IX Ver**
0.1384	0.3808	0.3599	0.4890	0.5765	0.3808
0.0416	0.4452	0.1796	0.0938	0.1796	0.9938
0.0938	0.5110	0.1262	0.1039	**0.0416**	0.9987

IX L	**IX R**	**X Ver**	**X L**	**X R**
0.0151	0.4890	0.3193	0.0967	**0.0039**
0.0603	0.0844	0.3193	0.9584	0.9681
0.2271	0.0603	0.3599	0.9723	0.9760

Table 3.2: p-value of one-side paired Wilcoxon test on Hausdorff distance. H_0: there is no significant improvement in the later processing step.

	CM	**I-V L**	**I-V R**	**VI Ver**	**VI L**
NL vs NLRF	0.7813	0.4375	1.0000	1.0000	0.6875
NL vs GC	1.0000	0.9805	0.8906	0.9598	0.4197
NLRF vs GC	1.0000	1.0000	0.8906	0.9539	0.4229

VI R	**VII Ver**	**VII Af L**	**VII At L**	**VII B L**	**VII Af R**
0.7500	1.0000	1.0000	0.6875	0.5000	0.9538
0.6523	0.9925	0.9994	0.9823	0.9966	0.9680
0.6523	0.9925	0.9988	0.9849	0.9976	0.9839

VII At R	**VII B R**	**VIII Ver**	**VIII L**	**VIII R**	**IX Ver**
0.2500	0.9375	1.0000	1.0000	0.4063	1.0000
0.2274	0.9829	0.4199	0.5391	0.9480	0.2344
0.3188	0.9473	0.0859	0.3711	0.9788	0.2188

IX L	**IX R**	**X Ver**	**X L**	**X R**
0.8125	0.5000	0.1563	0.7500	1.0000
0.9336	**0.0371**	0.2324	0.8389	0.9867
0.9453	**0.0371**	0.6250	0.8125	0.9867

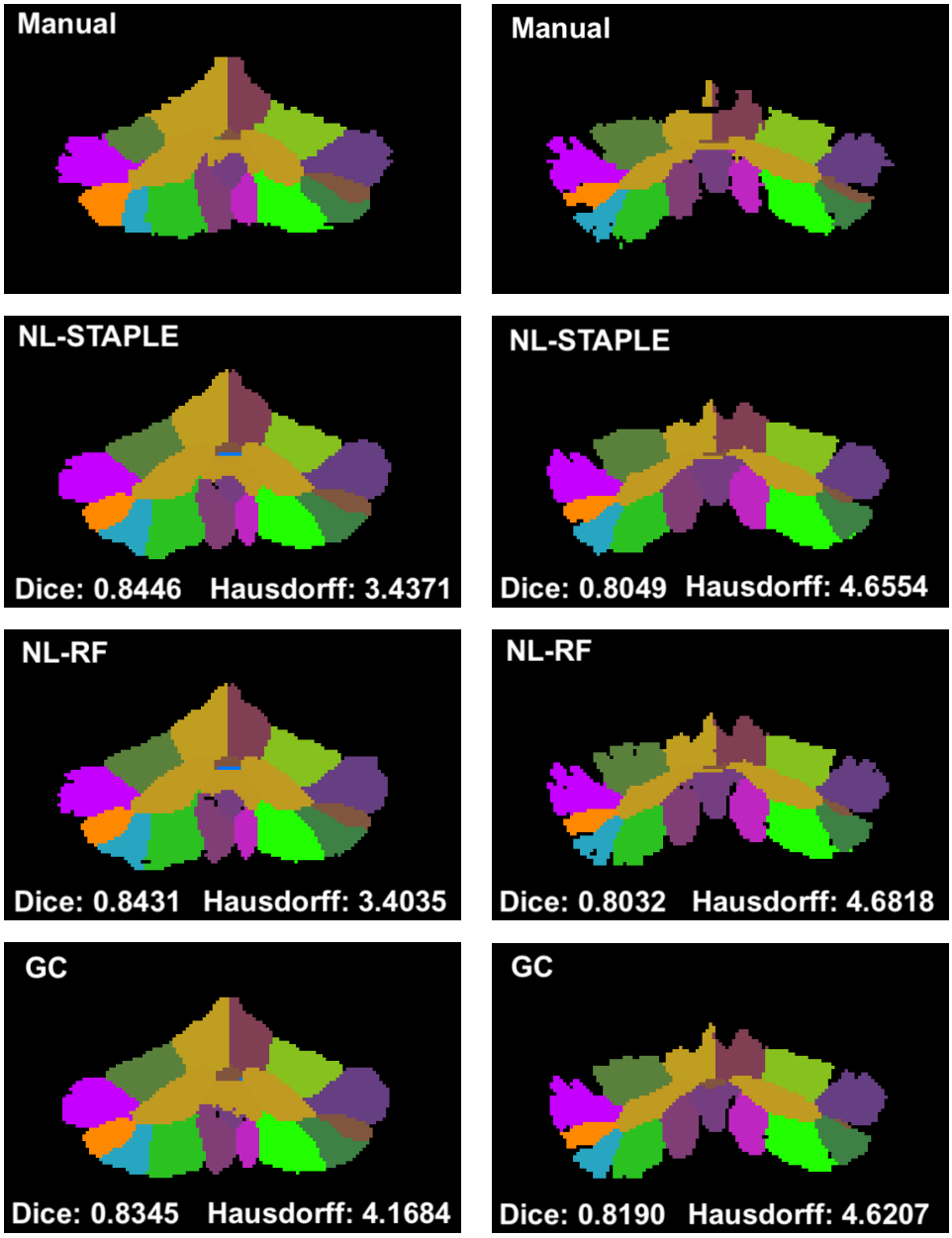


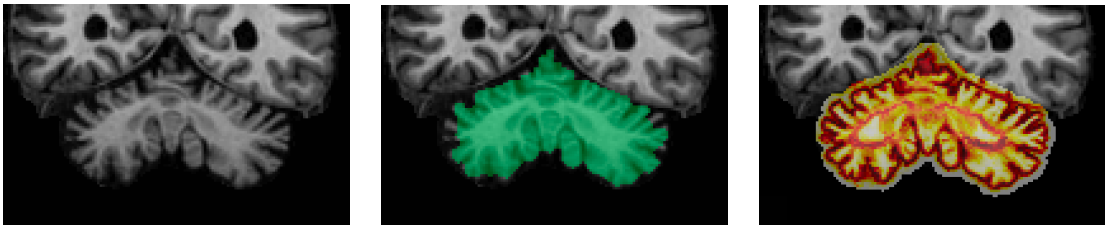
Figure 3.5: A visual comparison of segmentation results of **(a)** a healthy control and **(b)** a patient with SCA6.

3.2.3 Feature importance

In our segmentation scenario, the RF classifier plays an important role in both segmenting the whole cerebellum and finding cerebellar lobule boundaries. An RF classifier requires user defined features and corresponding true labels to do a voxel-wise classification. The performance of a random forest highly depends on the representability of features.

It has been accepted that an improper selection of features may lower the segmentation performance. For example, Trunk [46] points out that adding more but with less importance features does not necessarily guarantee a better classification performance. In our case, the random forest has 11 features, and we can study the inherent relationships of the 11 features both to validate the rationality of the software and to guide the potential improvement in the future.

Sensitivity analysis is a common tool to study the relevance and importance of feature variables [47]. Roughly speaking, sensitivity analysis assumes a feature to be redundant if there is no significant change in classification accuracy when perturbing the value of that feature. On the other hand, the importance of a feature can be estimated by perturbing a feature value and recording the decline in accuracy [48]. Following this idea, we did a leave-one-out experiment on the 15 subjects with expert delineations. The performance of two random forests were studied—RF for whole cerebellum segmentation and the RF for cerebellar lobule boundary classification. The task of an RF is to conduct a voxel-wise decision based on the feature vector and



(a) **(b)** **(c)**

Figure 3.6: **(a)** MR image of cerebellum. **(b)** Whole cerebellum segmentation. **(c)** Probability map of cerebellar lobule boundary.

a pretrained model. Specially, the whole cerebellum RF aims to decide whether a voxel belongs to cerebellar region or not, and the lobule boundary RF targets at differentiating outside cerebellum voxels, inside cerebellum voxels, and lobule boundary voxels. In each experiment, the average segmentation accuracy declines after perturbing a feature value was recorded. We use the accuracy decline as an estimation of feature importance. The average feature importance over all the 15 subjects is shown in Fig. 3.7. It is not superising to see that *intensity* plays a dominant role in finding lobule boundary regions, since lobule boundary has relative lower intensity values than cerebellar tissue regions, but higher intensity than background, which can be observed in Fig. 3.6. For classifying whole cerebellum region, *the distance to Corpus Medullare* contributes to more than 30% of the classification accuracy, which is also consistent with intuition since cerebellar lobules locate around the corpus medullare.

However, sensitivity analysis assumes independence of features variables, which is not true in most cases. Fig. 3.8 shows the correlation coefficient between the 11

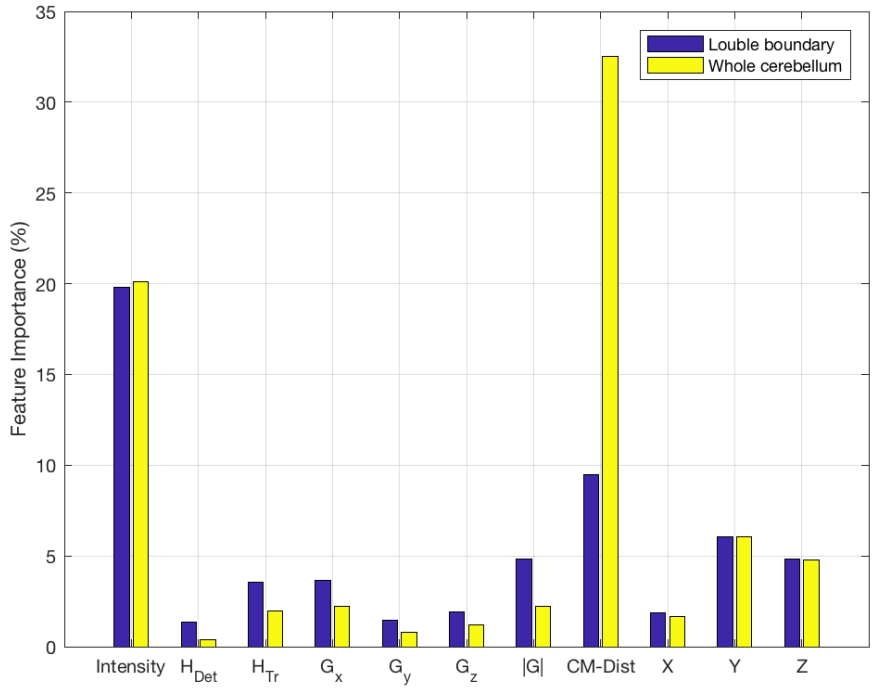


Figure 3.7: Average feature importance of cerebellar lobule boundary classification and whole cerebellum segmentation.

features. We can clearly see that there is correlation between features, therefore, Fig. 3.7 is just an approximate feature importance, and further analysis is needed to explore deeper relationship between features. From Fig. 3.8, we can also see that *intensity* has strong correlation with *CM-Dist*, which is consistent with the way that the features were extracted—in Yang et al.'s pipeline, the region of interest (ROI) is a 4-voxel binary dilation on a FreeSurfer result, where most non-cerebellar regions correspond to backgrounds. In such case, the farther a voxel away from the cerebellar region, the lower the intensity value is.

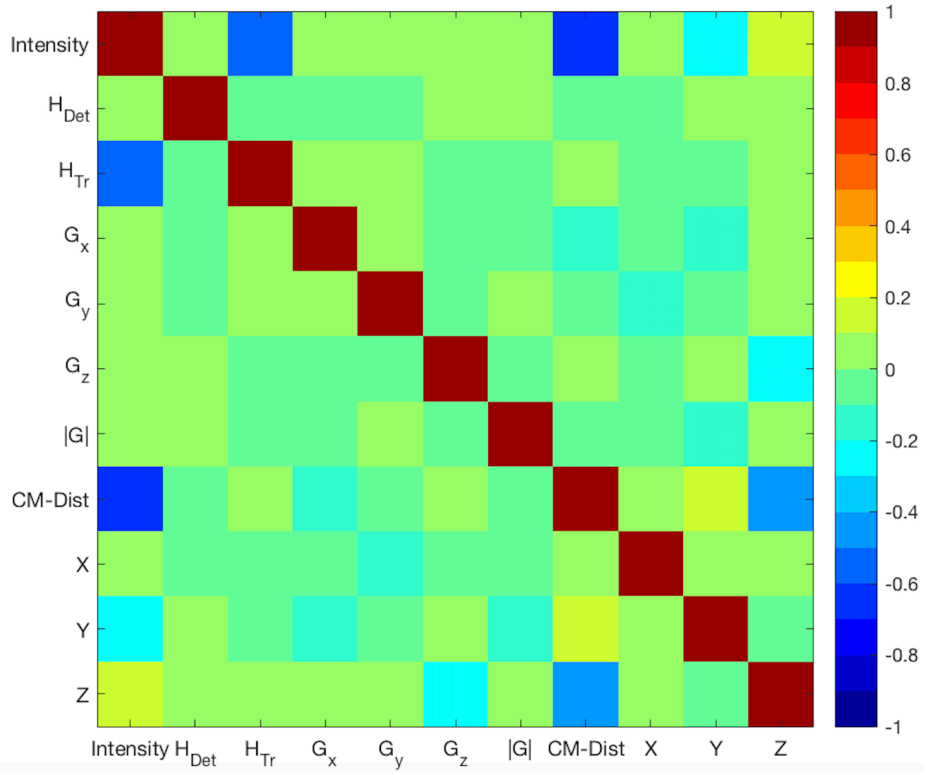


Figure 3.8: Pearson correlation coefficient of 11 features for RF classification.

3.2.4 Assessment of segmentation efficiency

The software was run on a computer with a 3.22 GHz CPU and a 251.7 GB RAM. The total computation time includes algorithm training and algorithm testing. The training procedure aims to find several RF models that can be used for RF testing. In testing, the runtime of four important steps was recorded: whole cerebellum segmentation using an RF, cerebellar lobule boundary classification using an RF, multi-atlas registration, and a GC segmentation step. As shown in Fig. 3.9, the software requires approximate 20 minutes to process one subject. Since the software was designed to

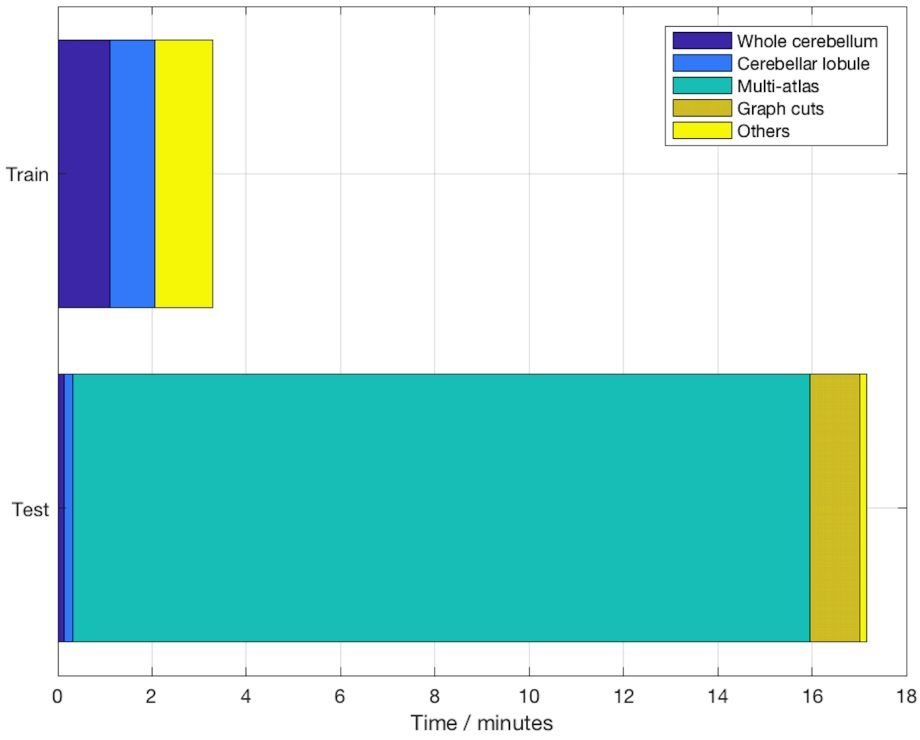


Figure 3.9: Running time of the cerebellum segmentation software.

be able to store pretrained RF models on the disk, in real applications the runtime can be reduced by 3 to 4 minutes since there is no need to retrain the RF models. In testing, the software spends most of its time doing multi-atlas registration. Compared with a typical whole brain parcellation approach, MaCRUISE [49], which also uses multi-atlas registration and requires several hours to run, the multi-atlas registration for cerebellum costs much less time due to the cerebellum image cropping at the very first step.

3.3 Categorizing failures

Oguz et al. [42] pointed out that the Dice coefficient is too simple to capture complicated system behaviors. In outlier detection, studying system behavior, especially the modes of failure, is of great importance. In this section, we categorize different kinds of segmentation failures. The failure categories are further used in Chapter 4 to guide outlier detection.

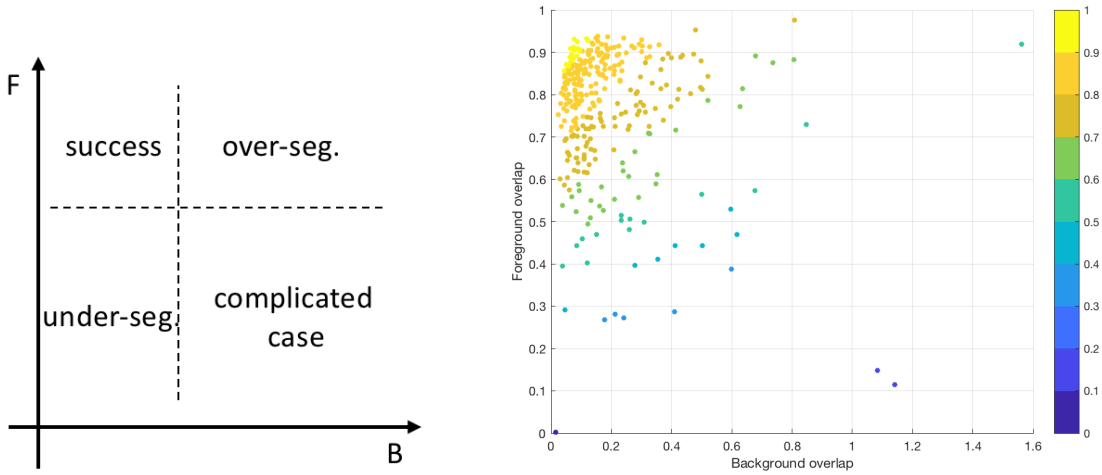
In our application, the segmentation result has 22 non-overlapping labels. If one label is segmented larger than ground truth, it will inevitably push other labels away. Therefore, we categorized the segmentation results into 4 groups: successful segmentation, under-segmentation (i.e., the segmentation is smaller than ground truth), over-segmentation (i.e., the segmentation is larger than ground truth), and a complicated case (i.e., shows characteristics of both over-segmentation and under-segmentation).

Now our goal is to find a proper metric that can capture all the four cases. We adopted and modified Udupa et al.'s idea [19] which uses NP and TP to evaluate segmentation results. We define two measurements F and B to be the overlap ratio with ground truth and the complement of ground truth, respectively:

$$F = \frac{|L \cap G|}{|G|}, \tag{3.1}$$

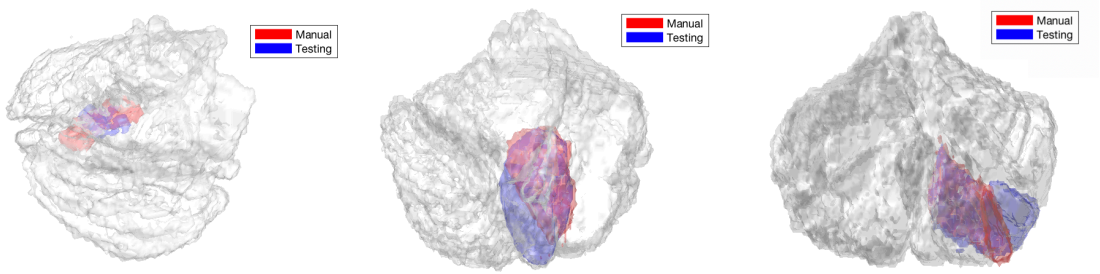
$$B = \frac{|L \cap \bar{G}|}{|G|}, \tag{3.2}$$

where L is a segmentation result, G is the segmentation ground truth, and $\bar{G}$ is the set complement of G. Obviously, a high F and low B together indicates a good segmenta-



(a) **(b)**

Figure 3.10: **(a)** An illustration and **(b)** a real plot of categorizing different segmentation cases in a 2D plot using defined parameters F and B. In **(b)**, the colorbar represents the Dice coefficient.



(a) **(b)** **(c)**

Figure 3.11: A visualization of three failure categories. **(a)** Under-segmentation. **(b)** Over-segmentation. **(c)** Complicated case.

tion. The other three segmentation categories can be represented by Fig. 3.10(a). Notice that under our definition, the metric B can have values greater than 1. Fig. 3.11 is a visualization of three failure categories, where the subjects shown were identified from Fig. 3.10(b).

3.4 Chapter summary

In this chapter, we introduced and validated Yang et al.'s cerebellar lobule segmentation software. Experiments show that the software can produce accurate segmentation results in most cases. In thin and small cerebellar structures, the software may fail to give a reliable result. In Section 3.3, we categorized different cases of failures, which assist us to do outlier detection in Chapter 4.

Chapter 4

Outlier Detection Using HMMs

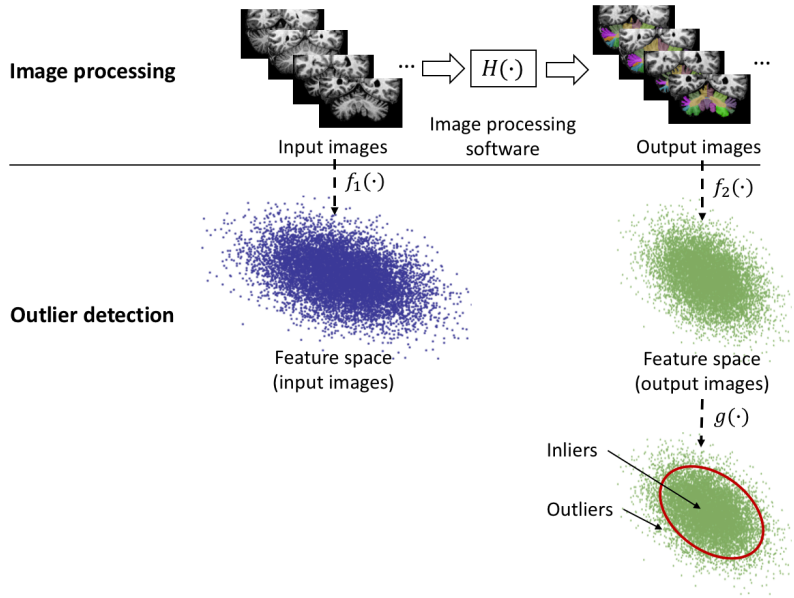
We introduce an outlier detection approach using Hidden Markov Models (HMMs) in this chapter. In Section 4.1, a general framework and the motivation of using HMMs for outlier detection are described. In Section 4.2, a delicate procedure of training HMM parameters using the Dice coefficient and a bag-of-words model is described. The experimental results of the proposed method are described in Section 4.3. The proposed method achieves both high sensitivity and high specificity.

4.1 General framework and motivations

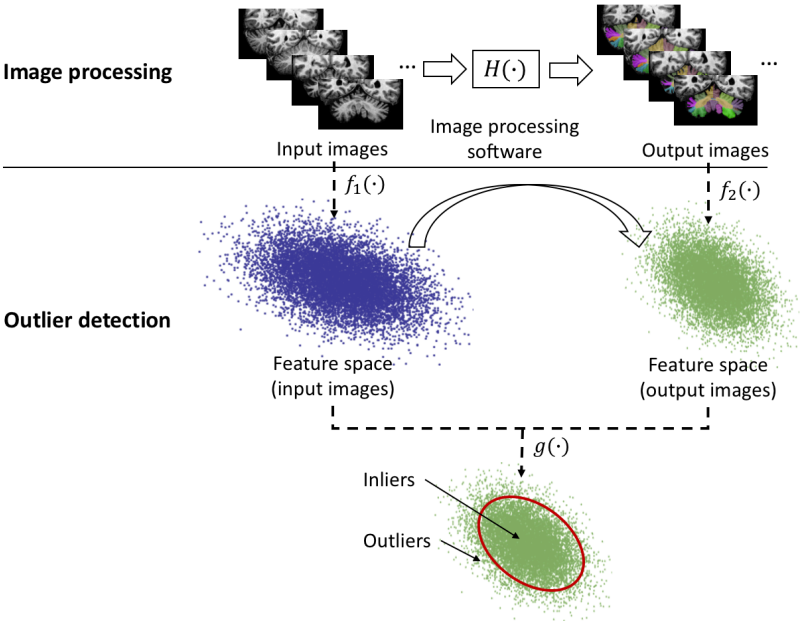
In medical imaging, many neuroimage processing pipelines [1,49,50] have multiple processing steps and outputs, which naturally inherit the characteristics of sequential data—the outputs are highly correlated with inputs. Inspired by this natural

property, we proposed an outlier detection framework using an HMM. In contrast to traditional outlier detection methods, which only focus on the output itself, our approach takes the input/output sequence into consideration. As mentioned in Chapter 2, the outlier detection task aims at finding a feature function $f(\cdot)$ and a decision function $g(\cdot)$. Assuming the images have already been mapped onto feature space by $f(\cdot)$, Fig. 4.1 shows an abstract comparison of traditional outlier detection method and the proposed method. Note that Fig. 4.1 only illustrates a one-step image processing software. For an image processing algorithm with multiple processing steps, the system function $H(\cdot)$ can be divided into multiple subfunctions, $H_i(\cdot)$, and the feature points are analyzed accordingly. Intuitively, in a single processing step, if the input is close to being an outlier, it is more likely for the corresponding output to be an outlier; the system (e.g., a single step of image processing software) behaves differently on different inputs in a feature space. Therefore, the superiority of proposed method is that, instead of simply treating outlier detection as an image classification problem, the proposed method captures all input/output information and analyzes the system behavior. This, to some extent, avoids the bottleneck problem mentioned in Section 2.2, and potentially improves the sensitivity and specificity of outlier detection.

Suppose a segmentation software has three steps before giving a final segmentation result, just like Yang et al.'s [1] approach, and suppose manual delineations are available for the HMM training stage. We denote the i^{th} processing step, its output,



(a)



(b)

Figure 4.1: An illustration of **(a)** traditional outlier detection approach, and **(b)** proposed method of an one-step image processing software.

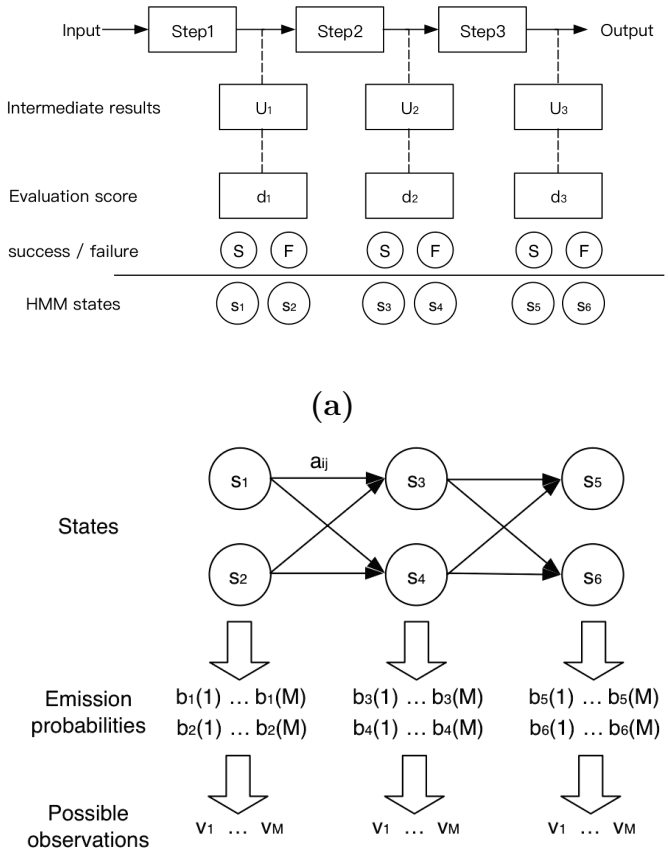


Figure 4.2: Framework of the proposed outlier detection using HMM: **(a)** An illustration of the segmentation software and **(b)** the established HMM.

and the corresponding evaluation score (e.g., a Dice coefficient) as A_i, U_i, and d_i. A 3-step framework of outlier detection using HMM is shown in Fig. 4.2. In the current work, we choose the evaluation score d_i to be the Dice coefficient. Notations of Fig. 4.2(b) are introduced with general HMM in Section 2.3.

We train the HMM using the intermediate and final results of Yang et al. [1], with a corresponding evaluation score d_i. For testing, we can only measure the observation sequence, $\boldsymbol{o}$, which is used in Viterbi's algorithm [51] to produce a possible state

sequence. Finally, the global assessment of segmentation result is obtained based on the state sequence.

4.2 Training HMMs

4.2.1 Training for transition probabilities

In this section, a method of training transition probabilities using the Dice coefficient is described. We also extend the idea of training the HMM and provide a connection between Dice coefficient and probability.

The simplest way to obtain a transition probability matrix $\boldsymbol{A}$ is to study the hidden state s_i and the following state s_j of each subject. This can be done by simply setting a threshold on the Dice coefficient and observing which subject fails in which state. Then use

$$\hat{a}_{ij} = P(s_j|s_i) \approx \frac{Count(s_i, s_j)}{Count(s_i)} \tag{4.1}$$

to get an estimate of a_{ij} [52], where $Count(s_i, s_j)$ means the number of subjects changing their state from s_i to s_j. However, due to the limited number of training samples in practice (i.e., the limited number of manual delineations) it is impractical to compute $P(s_j|s_i)$ based on the states s_i and s_j, because in this case, the method will have a large estimation error. Therefore, another method is needed to train the HMM with a limited number of training subjects.

Another promising way of getting transition probabilities is to use the Dice coefficient directly, since its value represents the segmentation's performance. We first note that Dice scores of complete failure, 0, and perfect segmentation, 1, already share similarity with the probability of a perfect segmentation. The states $s_1, \ldots, s_6$ (shown in Fig. 4.2) represent the success/failure of the different processing steps. Assuming Dice to be an approximation of being a perfect segmentation we have $d_i = P(s_{2i-1})$.

Next, we define an auxiliary variable y_i to be the contribution of the i^{th} step, as in

$$y_2 = \frac{1}{2}[(a_{13} + a_{23}) - (a_{14} + a_{24})] = (a_{13} + a_{23}) - 1, \tag{4.2}$$

$$y_3 = \frac{1}{2}[(a_{35} + a_{45}) - (a_{36} + a_{46})] = (a_{35} + a_{45}) - 1. \tag{4.3}$$

We note that y_i can have negative values, in which case the i^{th} step makes the result worse.

On one hand, the contribution score y_i can be understood as a positive effect (i.e., the ability of maintaining a successful segmentation, a_{13} or correcting a failed segmentation from a previous step, a_{23}) subtracted by a negative effect (i.e., the possibility of failing a successful segmentation, a_{14}, or the inability of correcting a failed segmentation, a_{24}). For example, if a processing step has $a_{13} = 1$ and $a_{23} = 0$, then y_2 should be zero. This is consistent with the fact that this processing step simply does nothing. Another example is that for an ideal processing step which can correct all the imperfection of a previous step, its corresponding a_{13} and a_{23} should both be 1. This will result in $y_2 = 1$, the maximum contribution. However, we can

also understand the contribution as the difference between the Dice scores of each step, that is $y_2 = d_2 - d_1$, and $y_3 = d_3 - d_2$.

The equations above give us one connection between the Dice coefficient and probability; another aspect comes from correlation. We define another auxiliary variable $\boldsymbol{r} = [r_{ij}]$ to be the Pearson correlation of d_i and d_j; then we have

$$r_{12} = \frac{1}{2}[(a_{13} + a_{24}) - (a_{23} + a_{14})] = a_{13} - a_{23}, \tag{4.4}$$

$$r_{23} = \frac{1}{2}[(a_{35} + a_{46}) - (a_{45} + a_{36})] = a_{35} - a_{45}. \tag{4.5}$$

To understand Eqs. 4.4 and 4.5, one can think that the correlation score between Dice is the ability of maintaining the previous state subtracted by the ability of inverting the previous state. By using the Equations above, the transition probability matrix $\boldsymbol{A}$ can be calculated based on Dice coefficient. Since the state change must follow the processing steps of the software, some entries of the transition probability matrix are manually set to zero. In our case, the transition probability matrix $\boldsymbol{A}$ can be written as

$$\begin{bmatrix} 0 & 0 & a_{13} & a_{14} & 0 & 0 \\ 0 & 0 & a_{23} & a_{24} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{35} & a_{36} \\ 0 & 0 & 0 & 0 & a_{45} & a_{46} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Furthermore, we can predict the Dice score of upcoming steps using the Dice score of the current step and the trained transition probabilities. From probability theory, we have

$$\hat{d}_2 = a_{13}d_1 + a_{23}(1 - d_1) \text{ and } \hat{d}_3 = a_{35}\hat{d}_2 + a_{45}(1 - \hat{d}_2). \tag{4.6}$$

In this section, we described how to train transition probabilities of an HMM. By connecting the Dice coefficient with probabilities, we are able to obtain transition probabilities in a limited number of training subjects. The validation results of the trained probabilities are described in Section 4.3.

4.2.2 Training for emission probabilities

In this section, the method of generating an observation sequence and training for the emission probability matrix $\boldsymbol{B}$ is introduced. The emission probability $b_i(k)$ is the probability of observing an observation v_k from state s_i. One of the key questions of this section is what to observe on each image processing step.

As mentioned before, QA using outlier detection is an *LP3* image classification task aiming at categorizing test images into successes and failures. Amongst all models for image classification, the "bag-of-words" (BoW) model has proved to have excellent performance in many application scenarios [53–55]. The core idea of BoW is to first represent a test image by a feature set, then use a a pretrained codebook to represent the feature set. Each entry of the codebook is called as a "visual word" [53]. The set of all "visual words" makes all possible HMM observations. Fi-

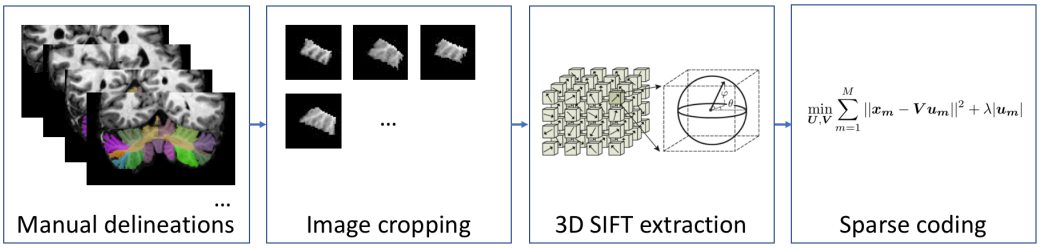


Figure 4.3: A flowchart of SIFT feature extraction and emission probability training.

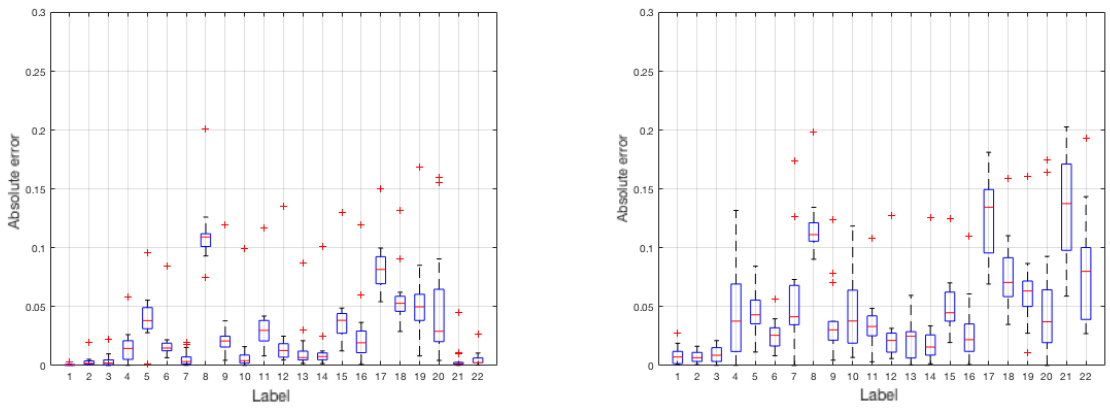
nally, the occurrence of each "visual word" is used as a frequency histogram to do image classification, which in our case, represents the emission probability.

Fig. 4.3 represents the procedure of training emission probabilities. First, the intensity images are cropped according to manual delineations. After image cropping, each cropped image represents a "true" structure to be segmented. Then SIFT features of each cropped image are extracted. Finally, we do a sparse coding based on BoW model both to reduce the feature dimension and generate HMM observations.

4.3 Outlier detection results

4.3.1 Validation of trained transition probabilities

To validate Eq. 4.6, we can use the trained transition probability matrix $\boldsymbol{A}$ to predict the Dice score of the upcoming processing steps. We have 15 patients which have manual delineations of each of the 22 labels. A leave-one-out experiment was done on the 15 patients. By selecting one patient as testing, we use the remaining 14 patients to train our transition probability matrix. To predict the Dice of upcoming



(a) **(b)**

Figure 4.4: Estimation error of **(a)** d_2 and **(b)** d_3 of the 22 labels in a leave-one-out experiment over 15 patients.

steps, we assume d_1 is the only known parameter of the testing image, and use Eq. 4.6 to obtain the estimated Dice $\hat{d}_2$ and $\hat{d}_3$. The absolute error in computing $\hat{d}_2$ and $\hat{d}_3$ is shown in Fig. 4.4. We can see that Eq. 4.6 achieves a high accuracy in predicting both $\hat{d}_2$ and $\hat{d}_3$, which validates the correctness of the trained transition probabilities and the proposed method. However, since $\hat{d}_3$ is estimated based on $\hat{d}_2$ instead of real d_2, the absolute error is larger. We note that this estimation is linear, thus when Dice has a nonlinear change the error is large.

4.3.2 Tests on 15 manual delineations

By studying the 15 patients with manual delineations, we found that the cerebellar segmentation software has good performance on most of the labels, i.e., the mean Dice of a label is higher than 0.75, which is consistent with the experiment result in [1].

Table 4.1: Confusion matrix of 15 patients which have manual delineations. The key code is: TPR - True Positive Rate; TNR - True Negative rate; FPR - False Positive Rate; FNR - False Negative Rate.

	TPR	**TNR**	**FPR**	**FNR**
Lobule VI - vermis	3/3	11/12	1/12	0/3
Lobule VIIB - left	3/4	8/11	3/11	1/4
Lobule VIIAt - right	3/3	11/12	1/12	0/3
Lobule VIIB - right	3/3	9/12	3/12	0/3

We then choose a threshold of 0.65 when comparing the manual delineations. Any score below this threshold is considered a segmentation failure; with this threshold we have 7 labels out of the 22 with a failure rate higher than 20%. The cerebellum labels with a relative high failure rate are the subdivisions of the vermis labelled as: Lobule VI, Lobule VIIAt, Lobule X; and the following lobules: left Lobule VIIB, right Lobule VIIB, left and right Lobule X. We then focused on the four largest labels with a failure rate higher than 20%, the largest structures were chosen to avoid the confound of unstable Dice scores on smaller structures. We conducted a leave-one-out experiment on these four labels. Table 4.1 shows the confusion matrix of our leave-one-out experiment on the four labels of 15 patients. The reported truth was selected based on the Dice of the final segmentation. We define a predicted failure whose truth is also a failure as true positive. We notice that the proposed method achieves both high true positive rate (TPR) and a high true negative rate (TNR).

4.3.3 Experiment on more datasets

We ran the proposed outlier detection software on the Tomacco dataset. There are 46 subjects in the Tomacco dataset: 16 healthy controls, 6 patiences with SCA6 and 24 subjects with other cerebellar related disease. Note that the 15 subjects with manual delineations come from the Tomacco dataset, and since they have already been analyzed in Section 4.3.2, in this section, we exclude the 15 subjects and did the experiment on the remaining 31 subjects. Of the 31 subjects, there are 10 healthy controls, and 21 subjects with cerebellar related disease.

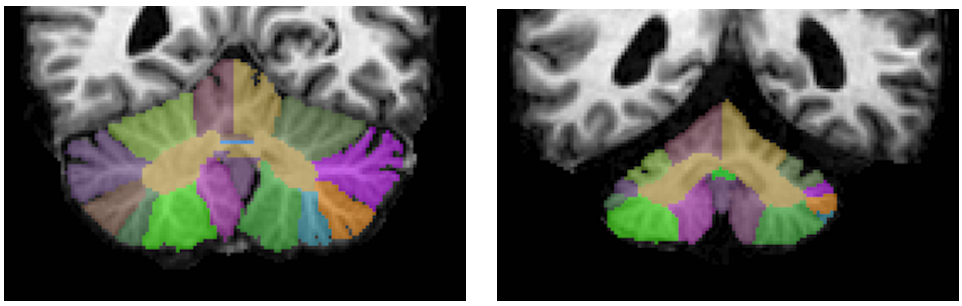
Yang et al.'s [1] cerebellar lobule segmentation software was successfully trained using the 15 manual delineations and was run on the remaining 31 subjects. Three intermediate outputs including the final output were processed and analyzed by our proposed outlier detection software. The outlier detection software was trained using real segmentation failures from a leave-one-out experiment on the 15 subjects. Due to the limited number of training data and the small size of the cerebellar lobules, we only experimented on the following 4 lobules: Lobule VI - vermis, Lobule VIIB - left, Lobule VIIB – right, and Lobule VIIAt - left. The QA result is shown in Table 4.2. The 31 subjects were indexed from 1 to 31.

Since we do not have manual delineations of the 31 subjects, it is impractical to quantitatively verify the outlier detection result. So we provide a visualization of segmentation failures and some sample subjects which have manual delineations. From Table 4.2, we found that three labels out of four of Subject #11 were detected

Table 4.2: Outlier detection results of 31 subjects in Tomacco dataset.

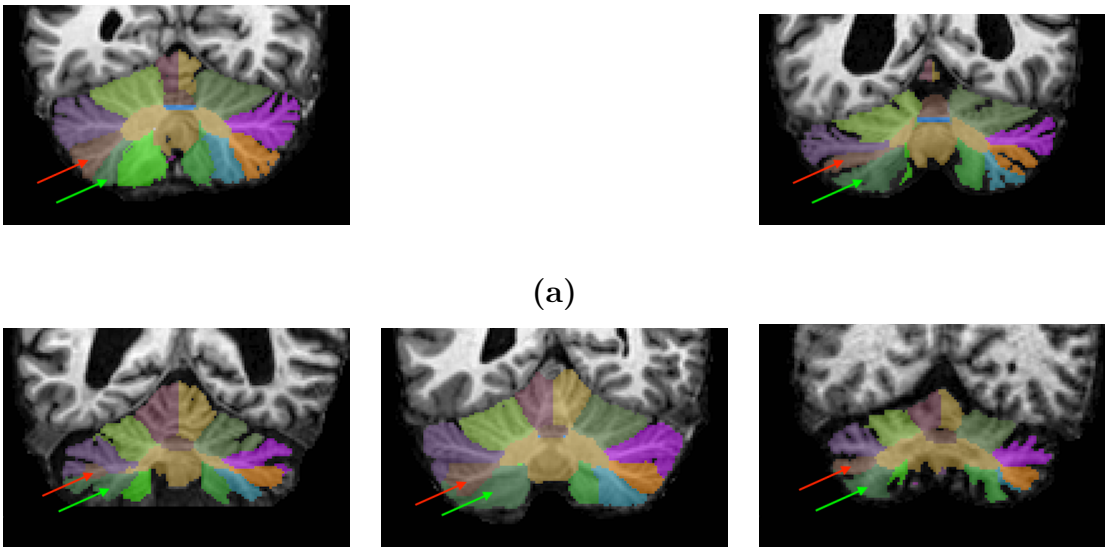
	# of detected outliers	**Subject index**
Lobule VI - vermis	4/31	18, 25, 26, 28
Lobule VIIAt - left	7/31	11, 14, 15, 19, 20, 22, 23
Lobule VIIB - left	7/31	3, 7, 9, 11, 20, 22, 23
Lobule VIIB - right	5/31	10, 11, 13, 26, 29

as segmentation failures. Fig. 4.5 is a visualization of Subject #11 and a healthy subject with manual delineation. We observe that Subject #11 has very abnormal cerebellar shape, which makes it distinct from other subjects in many cerebellar lobules. Another observation is that both Lobule VIIAt left and Lobule VIIB left of Subject #20, 22, and 23 were detected as segmentation failures, although each label was analyzed independently. We visualize the two lobules of the three subjects together with a manually delineated subject in Fig. 4.6. From the figure, we notice that the two lobules are touching each other, which means if one lobule were wrongly segmented, the other would probably be a segmentation failure as well.



(a) **(b)**

Figure 4.5: A visual comparison of Subject #11 of Tomacco dataset. **(a)** A healthy subject with manual delineation. **(b)** Tomacco Subject #11.



(a)

(b)

Figure 4.6: A visualization of Lobule VIIAt - left (red arrow) and Lobule VIIB - left (green arrow). **(a)** Manual delineations of a healthy control (left), and a subject with SCA6 (right). **(b)** Automatic segmentation results. From left to right: Subject #20, 22, and 23.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we evaluated a segmentation software for human cerebellar lobule parcellation. We then presented a new QA approach using outlier detection for the segmentation software. In the first part of the thesis, we formulated the application domain of our QA task, and introduced several metrics of evaluating a segmentation software. We then evaluated the cerebellar lobule segmentation software in terms of its accuracy and efficiency using the previously mentioned metrics. The work provides a systematic and statistical approach of evaluating a segmentation software in medical image analysis.

In the second part of the thesis, we introduced our outlier detection software using an HMM. The successful combination of the HMM and the Bag-of-Words model

enables the outlier detection software a global assessment on the segmentation results instead of only looking at the final result itself. Experiments were done on real datasets. Quantitative analysis on 15 subjects with manual delineations show that the outlier detection software achieves both high specificity and high sensitivity. The QA results on more data give researchers a clue about when and where the segmentation software might fail in real segmentation scenarios.

5.2 Future work

In this thesis, we proposed an automatic QA method using outlier detection for a cerebellar lobule segmentation software. The proposed method moves one step forward than traditional QA methods which only focus on the final results. However, this work is certainly not an ending point, and it has many improvements that can be done in future work, as described next:

First, in this work, we did QA on each segmentation label independently. However, in real scenario where more than two (foreground and background) non-overlapping labels are segmented, there is correlation between labels, especially between adjacent labels. To be specific, an over-segmented label is likely to make its adjacent labels under-segmented. Therefore, a promising direction of future work is to study the relationship between segmentation labels, which may potentially avoid some false alarms and guarantee a higher sensitivity.

Second, the feature selection work of the proposed method is not sufficiently generous. Experiments show that an improper selection of features may degrade the performance of a QA. In this work, the SIFT feature was manually and empirically selected and they may not work well in other segmentation scenarios. Therefore, an automatic and generous feature selection procedure is highly desirable. In recent work, several researchers have successfully trained autoencoders using deep neural networks. The autoencoders learn discriminative features that can be used for image reconstruction, denoising, and even shape correction. The successful implementation of autoencoders lights the way for more reliable and discriminative feature selection methods.

Third, the proposed method does not handle intrinsic variations very well. For example, subjects with disease and healthy controls have distinct anatomical structures. Ignoring these variations may cause false alarms (i.e., a successful segmentation but is treated as a failure) and misclassifications (i.e., a segmentation failure but is not detected) in outlier detection. In future work, researchers should consider classifying subcategory before conducting a QA. In addition, it would be also interesting to explore and rank the effects of different variations, such as healthy/disease, young/old, male/female, and success/failures.

Bibliography

[1] Z. Yang, C. Ye, J. A. Bogovic, A. Carass, B. M. Jedynak, S. H. Ying, and J. L. Prince, "Automated cerebellar lobule segmentation with application to cerebellar structural analysis in cerebellar disease," *NeuroImage*, vol. 127, pp. 435–444, 2016.

[2] H. C. Leiner, A. L. Leiner, and R. S. Dow, "Cognitive and language functions of the human cerebellum," *Trends in neurosciences*, vol. 16, no. 11, pp. 444–447, 1993.

[3] J. D. Schmahmann and J. C. Sherman, "The cerebellar cognitive affective syndrome." *Brain: a journal of neurology*, vol. 121, no. 4, pp. 561–579, 1998.

[4] C. J. Stoodley and J. D. Schmahmann, "Functional topography in the human cerebellum: a meta-analysis of neuroimaging studies," *Neuroimage*, vol. 44, no. 2, pp. 489–501, 2009.

[5] T. Wu and M. Hallett, "The cerebellum in parkinson s disease," *Brain*, vol. 136, no. 3, pp. 696–709, 2013.

[6] S. Standring, *Gray's anatomy e-book: the anatomical basis of clinical practice*. Elsevier Health Sciences, 2015.

[7] J. D. Schmahmann, J. Doyon, D. McDonald, C. Holmes, K. Lavoie, A. S. Hurwitz, N. Kabani, A. Toga, A. Evans, and M. Petrides, "Three-dimensional mri atlas of the human cerebellum in proportional stereotaxic space," *Neuroimage*, vol. 10, no. 3, pp. 233–260, 1999.

[8] J. Diedrichsen, "A spatially unbiased atlas template of the human cerebellum," *Neuroimage*, vol. 33, no. 1, pp. 127–138, 2006.

[9] J. A. Bogovic, P.-L. Bazin, S. H. Ying, and J. L. Prince, "Automated segmentation of the cerebellar lobules using boundary specific classification and evolution," in *International Conference on Information Processing in Medical Imaging*. Springer, 2013, pp. 62–73.

[10] J. A. Bogovic, B. Jedynak, R. Rigg, A. Du, B. A. Landman, J. L. Prince, and S. H. Ying, "Approaching expert results using a hierarchical cerebellum parcellation protocol for multiple inexpert human raters," *Neuroimage*, vol. 64, pp. 616–629, 2013.

[11] C. E. Willis, "Quality assurance for medical imaging," in *Practical Imaging Informatics*. Springer, 2009, pp. 197–211.

[12] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[13] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[14] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Biol. Skr.*, vol. 5, pp. 1–34, 1948.

[15] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. Van Der Kouwe, R. Killiany, D. Kennedy, S. Klaveness *et al.*, "Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain," *Neuron*, vol. 33, no. 3, pp. 341–355, 2002.

[16] J. Diedrichsen, J. H. Balsters, J. Flavell, E. Cussans, and N. Ramnani, "A probabilistic mr atlas of the human cerebellum," *Neuroimage*, vol. 46, no. 1, pp. 39–46, 2009.

[17] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[18] J. E. Romero, P. Coupé, R. Giraud, V.-T. Ta, V. Fonov, M. T. M. Park, M. M.

Chakravarty, A. N. Voineskos, and J. V. Manjón, "Ceres: A new cerebellum lobule segmentation method," *NeuroImage*, vol. 147, pp. 916–924, 2017.

[19] J. K. Udupa, V. R. Leblanc, Y. Zhuge, C. Imielinska, H. Schmidt, L. M. Currie, B. E. Hirsch, and J. Woodburn, "A framework for evaluating image segmentation algorithms," *Computerized Medical Imaging and Graphics*, vol. 30, no. 2, pp. 75–87, 2006.

[20] K. Li, "Quality assurance using outlier detection for automatic segmentation of cerebellar peduncles," Ph.D. dissertation, Johns Hopkins University, 2015.

[21] Z. Yang, S. M. Abulnaga, A. Carass, K. Kansal, B. M. Jedynak, C. Onyike, S. H. Ying, and J. L. Prince, "Landmark based shape analysis for cerebellar ataxia classification and cerebellar atrophy pattern visualization," in *Medical Imaging 2016: Image Processing*, vol. 9784. International Society for Optics and Photonics, 2016, p. 97840P.

[22] K. Li, C. Ye, Z. Yang, A. Carass, S. H. Ying, and J. L. Prince, "Quality assurance using outlier detection on an automatic segmentation method for the cerebellar peduncles," in *Medical Imaging 2016: Image Processing*, vol. 9784. International Society for Optics and Photonics, 2016, p. 97841H.

[23] S.-G. Miaou and S.-T. Chen, "Automatic quality control for wavelet-based compression of volumetric medical images using distortion-constrained adaptive vec-

tor quantization," *IEEE transactions on medical imaging*, vol. 23, no. 11, pp. 1417–1429, 2004.

[24] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[25] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[27] S. Z. Li, "Markov random field models in computer vision," in *European conference on computer vision*. Springer, 1994, pp. 361–370.

[28] C. Wachinger, A. Rieckmann, and M. Reuter, "Latent processes governing neuroanatomical change in aging and dementia," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 30–37.

[29] E. Varol, A. Sotiras, and C. Davatzikos, "Structured outlier detection in neuroimaging studies with minimal convex polytopes," in *International Conference*

on Medical Image Computing and Computer-Assisted Intervention. Springer, 2016, pp. 300–307.

[30] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[31] B. Sirmacek and C. Unsalan, "Urban-area and building detection using sift keypoints and graph theory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1156–1167, 2009.

[32] C.-C. Wang and K.-C. Wang, "Hand posture recognition using adaboost with sift for human robot interaction," in *Recent progress in robotics: viable robotic service to human*. Springer, 2007, pp. 317–329.

[33] M. Toews, W. Wells III, D. L. Collins, and T. Arbel, "Feature-based morphometry: Discovering group-related anatomical patterns," *NeuroImage*, vol. 49, no. 3, pp. 2318–2327, 2010.

[34] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European conference on computer vision*. Springer, 2002, pp. 128–142.

[35] A. C. Evans, S. Marrett, P. Neelin, L. Collins, K. Worsley, W. Dai, S. Milot, E. Meyer, and D. Bub, "Anatomical mapping of functional activation in stereotactic coordinate space," *Neuroimage*, vol. 1, no. 1, pp. 43–53, 1992.

[36] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[37] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006.

[38] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[39] A. J. Asman and B. A. Landman, "Non-local statistical label fusion for multi-atlas segmentation," *Medical image analysis*, vol. 17, no. 2, pp. 194–208, 2013.

[40] C. Zhao, A. Carass, J. Lee, Y. He, and J. L. Prince, "Whole brain segmentation and labeling from ct using synthetic mr images," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2017, pp. 291–298.

[41] J. Glaister, A. Carass, T. NessAiver, J. V. Stough, S. Saidha, P. A. Calabresi, and J. L. Prince, "Thalamus segmentation using multi-modal feature classification: Validation and pilot study of an age-matched cohort," *NeuroImage*, vol. 158, pp. 430–440, 2017.

[42] I. Oguz, A. Carass, D. L. Pham, S. Roy, N. Subbana, P. A. Calabresi, P. A. Yushkevich, R. T. Shinohara, and J. L. Prince, "Dice overlap measures for objects of unknown number: Application to lesion segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 3–14.

[43] T. Heimann, B. Van Ginneken, M. A. Styner, Y. Arzhaeva, V. Aurich, C. Bauer, A. Beck, C. Becker, R. Beichel, G. Bekes *et al.*, "Comparison and evaluation of methods for liver segmentation from ct datasets," *IEEE transactions on medical imaging*, vol. 28, no. 8, pp. 1251–1265, 2009.

[44] V. Yeghiazaryan and I. Voiculescu, "An overview of current evaluation methods used in medical image segmentation," Tech. Rep. CS-RR-15-08, Department of Computer Science, University of Oxford, Oxford, UK, Tech. Rep., 2015.

[45] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[46] G. V. Trunk, "A problem of dimensionality: A simple example," *IEEE Transactions on pattern analysis and machine intelligence*, no. 3, pp. 306–307, 1979.

[47] B. Iooss and P. Lemaître, "A review on global sensitivity analysis methods," in *Uncertainty management in simulation-optimization of complex systems*. Springer, 2015, pp. 101–122.

[48] A. Razmjoo, P. Xanthopoulos, and Q. P. Zheng, "Online feature importance ranking based on sensitivity analysis," *Expert Systems with Applications*, vol. 85, pp. 397–406, 2017.

[49] Y. Huo, A. J. Plassard, A. Carass, S. M. Resnick, D. L. Pham, J. L. Prince,

and B. A. Landman, "Consistent cortical reconstruction and multi-atlas brain segmentation," *NeuroImage*, vol. 138, pp. 197–210, 2016.

[50] K. Gorgolewski, C. D. Burns, C. Madison, D. Clark, Y. O. Halchenko, M. L. Waskom, and S. S. Ghosh, "Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python," *Frontiers in neuroinformatics*, vol. 5, p. 13, 2011.

[51] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[52] P. Blunsom, "Hidden markov models," 2004.

[53] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang, "Multi-label sparse coding for automatic image annotation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 1643–1650.

[54] J. Wu and J. M. Rehg, "Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 630–637.

[55] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on.* IEEE, 2007, pp. 1–8.

Vita

Lianrui Zuo was born in March 1994, Liaoning Province, China. He received a Bachelor of Engineering degree in Instrument Science and Technology at Jilin University, China in June 2016. He then enrolled the Master of Science in Engineering (M.S.E) program in Electrical and Computer Engineering at Johns Hopkins University in August 2016. He conducted his research in the Image Analysis and Communications Lab under the direction of Dr. Jerry L. Prince. His research interests include automatic quality assurance, statistical shape analysis, machine learning, and feature selection.

Vita

Lianrui Zuo was born in March 1994, Liaoning Province, China. He received a Bachelor of Engineering degree in Instrument Science and Technology at Jilin University, China in June 2016. He then enrolled the Master of Science in Engineering (M.S.E) program in Electrical and Computer Engineering at Johns Hopkins University in August 2016. He conducted his research in the Image Analysis and Communications Lab under the direction of Dr. Jerry L. Prince. His research interests include automatic quality assurance, statistical shape analysis, machine learning, and feature selection.